RADC-TR-82-18
Final Technical Report
February 1982

PATTERN RECOGNITION MONOGRAPH I

PAR Technology Corporation

Thomas V. Edwards

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-82-18 has been reviewed and is approved for publication.

APPROVED: Haywood E. Webb, Jr.
HAYWOOD E. WEBB, Jr.
Project Engineer

APPROVED:
JOHN J. MARCINIAK, Colonel, USAF
Chief, Command and Control Division

FOR THE COMMANDER: John P. Huss
JOHN P. HUSS
Acting Chief, Plans Office

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-82-18	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) PATTERN RECOGNITION MONOGRAPH I		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report
		6. PERFORMING ORG. REPORT NUMBER PAR Report #80-52
7. AUTHOR(s) Thomas V. Edwards		8. CONTRACT OR GRANT NUMBER(s) F30602-80-C-0131
9. PERFORMING ORGANIZATION NAME AND ADDRESS PAR Technology Corporation Seneca Plaza, Rt 5 New Hartford NY 13413		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS J.O. 55812311
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (COTD) Griffiss AFB NY 13441		12. REPORT DATE February 1982
		13. NUMBER OF PAGES 152
14. MONITORING AGENCY NAME & ADDRESS (*if different from Controlling Office*) Same		15. SECURITY CLASS. (*of this report*) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Haywood Webb (COTD)

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Pattern Recognition
Interactive Computer System

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This report describes a series of interactive computer driven display systems for aids in the solution of pattern recognition problems. Comparative analyses are given for each system from a software, and user point of view.

5. p 1-1

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

TABLE OF CONTENTS

LIST OF FIGURES

SECTION 1

INTRODUCTION

The classical approach to the empirical solution of a waveform pattern recognition problem invariably involves repeated iteration between feature definition, feature extraction, and feature evaluation. Real world pattern recognition problems of this type with little or no "a priori" information available indicate the need for an interactive, graphics oriented feature definition/evaluation system. Because no single feature extraction algorithm satisfies all waveform pattern recognition problem requirements, it is desirable to provide a wide variety of flexible techniques for data manipulation and transformation. The inspection of results from specified data manipulations and transformations must be presented in such a way to exploit the human's ability to identify candidate waveform features. Just as there is no single algorithm to satisfy all feature extraction requirements, there is no single algorithm to evaluate the merit of all data transformation results. As shown in Figure 1-1, both the feature definition and evaluation phase of the waveform pattern recognition are interactively based. Effective graphic and statistical presentation of transformation results by the system and their interpretation by a human operator together with his "a priori" knowledge of the data and the nature of the problem are the key to the determination of a successful solution to the problem.

Keywords: IDRS (Interactive Digital Processor Simulator). (KR)

Perhaps the single most critical phase of the overall solution of a waveform pattern recognition problem is the feature definition. No

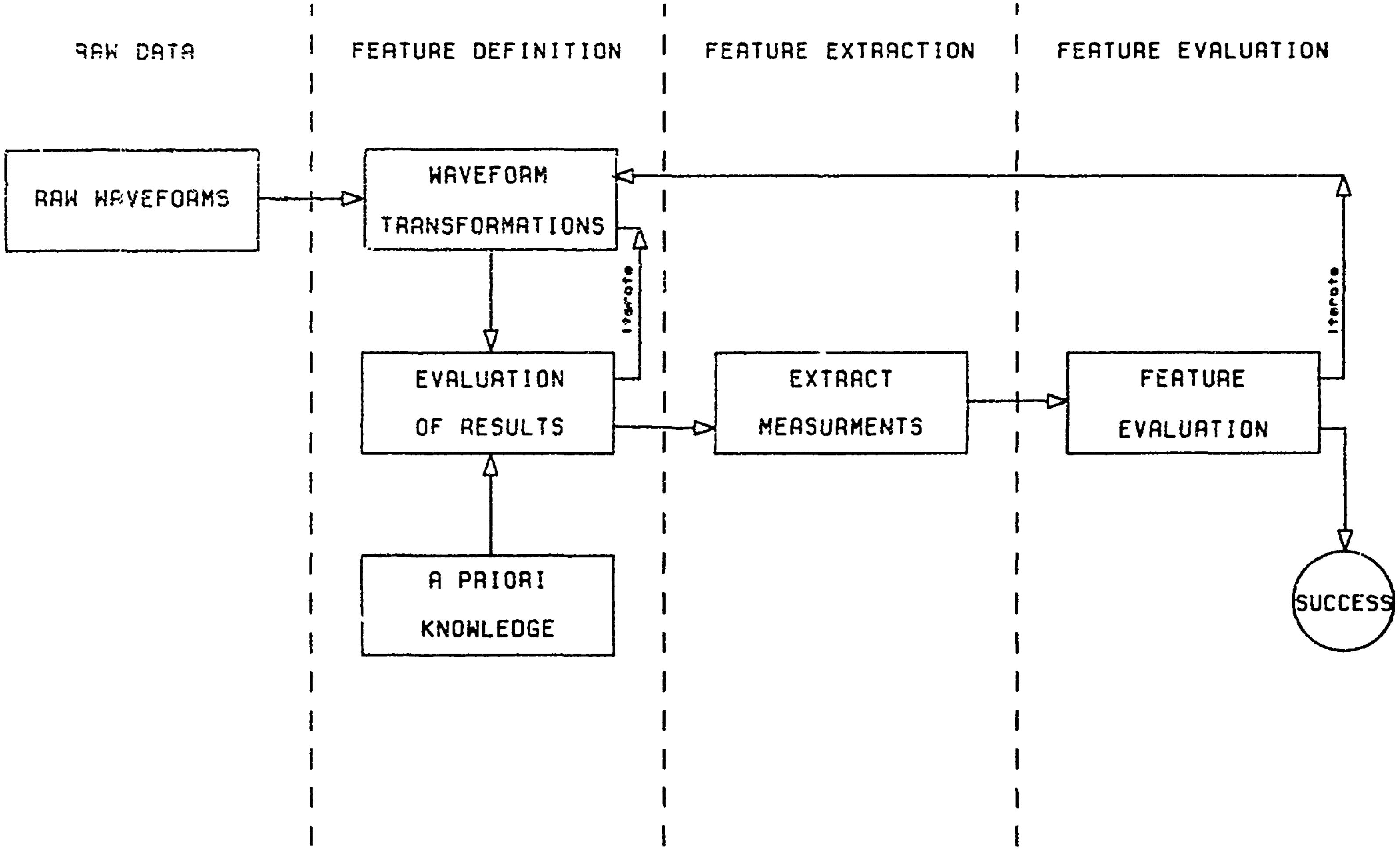


Figure 1.1 Phases of the solution to pattern recognition problems

matter how precise the feature extraction algorithms, or how powerful the feature evaluation mechanism, indiscriminate feature definition cannot be completely compensated for. This fact necessitates the use of an effective interface to a software system which provides man with ample tools, so as not to excessively restrict the realm and scope of the hypotheses which may be considered as possible problem solutions. The Interactive Digital Receiver Simulator (IDRS) was developed to fulfil the requirement for a versatile and effective interactive feature definition system.

IDRS was developed for RADC as a task under contract F19628-76-C-0002 and currently supports the RADC Pattern Recognition Design Facility. IDRS has its roots with the Long Waves Processing System (LPS), and was developed to be a front-end to the Waveform Processing System (WPS). The system is, however, quite versatile and is self-sufficient for many waveform analysis applications. The flexibility of the system comes from its "work bench" style design, that is, the user has the ability to assemble receiver modules as deemed appropriate and investigate the waveform data between modules through effective graphic displays. The user is provided with both time and frequency analysis capabilities including spectral waveform weighting and averaging.

The application of IDRS to waveform analysis problems in the past has proven highly effective. Although the nature of feature definition problems has continuously grown more sophisticated and complex, the

flexibility of the IDRS has enabled its continued usefulness. The remainder of this section presents a macroscopic description of IDRS and then discusses IDRS as a pattern recognition tool.

1.1. MACROSCOPIC DESCRIPTION OF IDRS SOFTWARE

IDRS is a collection of waveform transformation modules (or tools) which, together with data manipulation capabilities, form a digital receiver work bench. Effective solutions to waveform pattern recognition problems often include the formation of a hypothesis describing how the data was generated. This assumption leads the user to further hypotheses concerning the location of potential features in the data. The IDRS provides a means for the user to tailor a system to his data in an effort to investigate such hypothesis. The investigation of potential features may range from as simple as waveform spectral analysis, or as complex as data clock stability derived from a receiver IF signal.

Considering the wide variety of system capabilities, how knowledgeable must the user be to use IDRS? Because of the effective man-machine interface, which provides for data manipulation and graphic presentation of results, the user requirements for IDRS can best be identified by the scope of the particular problem. If the user can understand the nature and scope of the problem he is dealing with, he has enough background to use the system. This is to say, the student interested in harmonic analysis of speech data should have no major problems; however, the same student's talents are likely to be inadequate for effectively solving a complex waveform pattern recognition problem.

1.1.1. IDRS: The Receiver

IDRS allows a user to interactively generate and display signal transformation patterns (and, optimally, a hardcopy library of these displays) for detailed human analysis and, in addition to, format pre-processed waveform segments for input to the WPS for feature extraction. In so doing, the IDRS user interactively creates a digital receiver to his specification. Figure 1-2 illustrates the signal processing capabilities of IDRS. Inputs to the system are in the form of digitized waveforms. The selection of a portion of the waveform for further in-depth analysis is termed "Segmentation of the Signal." A particularly attractive feature of IDRS is its ability to permit identification of any sample point of the waveform, or of a waveform segment which starts at a particular sample point. The system aids the user in selecting a waveform segment by selectively displaying requested segments for inspection as a function of both time and frequency.

Once a waveform segment is identified, the user can begin an in-depth investigation. The investigation need not utilize the complete receiver shown in Figure 1-2. Any of the modules may be used independently, or omitted. The receiver can be built interactively, monitoring the input and output of each module as the design proceeds. Alternately, the entire receiver configuration can be specified immediately. The following paragraphs illustrate a general digital receiver simulation using the IDRS.

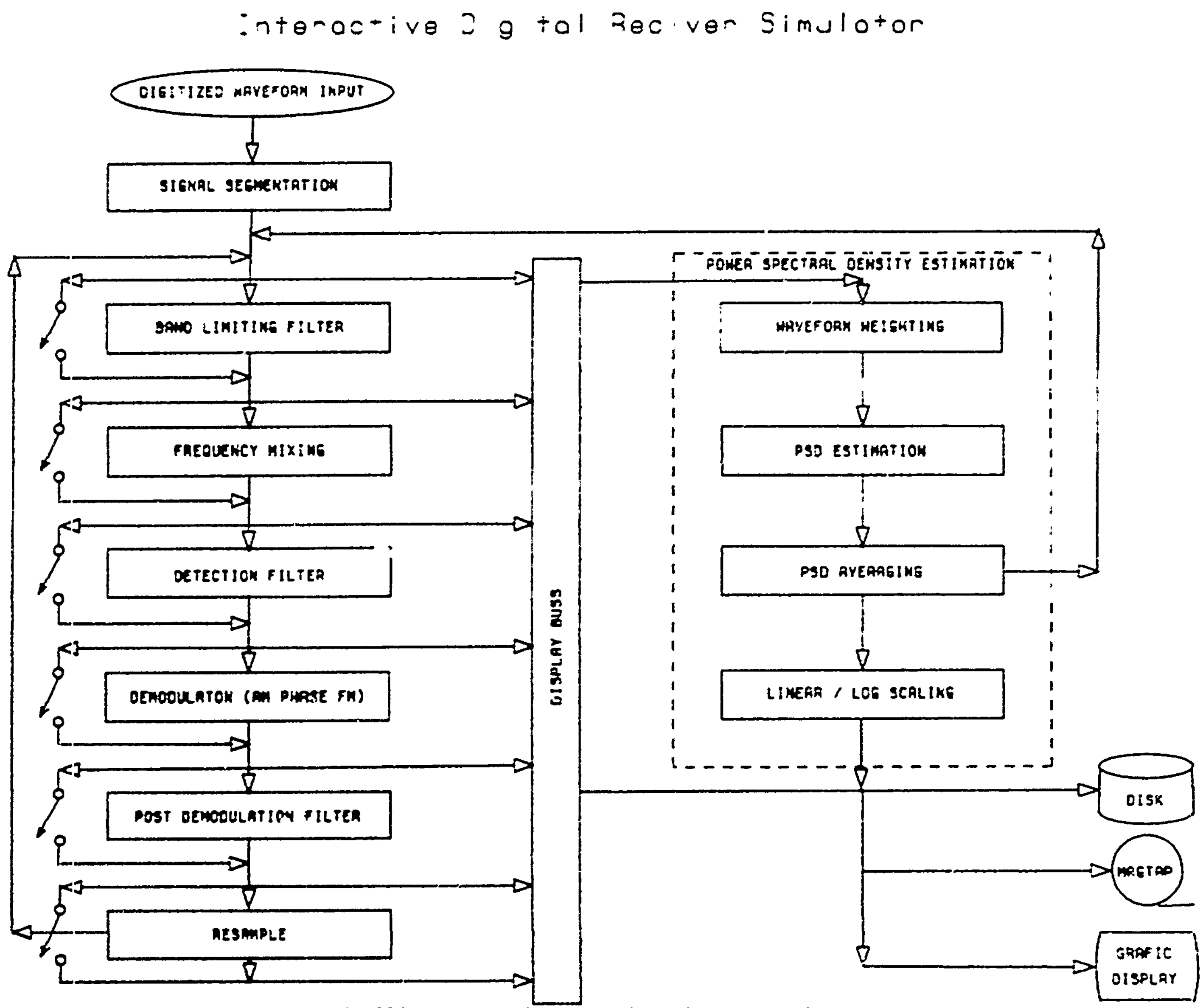


Figure 1.2 Block diagram of IDRS showing the controling logic switche

The first phase of the simulation is the spectral analysis of a representative waveform segment. After determining the desired signal bandwidth, the predetection filter characteristic is selected. The filter is specified in terms of filter type (Butterworth or Chebyshev), center frequency, bandwidth and roll-off. The filter characteristic is selected to best reject noise and interference (band limit the signal) and also as a first step in obtaining greater frequency resolution in PSD estimations.

Frequency resolution in spectral analysis is related to the duration of the signal segment that is Fourier transformed. For digital spectral analysis, the length of the time segment is fixed by the sampling rate and the size of the Fast Fourier Transform (FFT). The options for increasing the frequency resolution in digital spectral analysis are to increase the number of samples transformed (which is usually limited from software considerations), or decrease the sample rate. IDRS has a maximum size FFT of 1024 complex points. To increase FFT frequency resolution, the signal data is band limited through filtering, selecting only the portion of the signal spectrum over which higher resolution is desired. This filtering is usually accomplished with the predetection filter. Once the signal has been band limited, it can be resampled to achieve a lower effective sample rate. For example, if the signal is band limited by filtering to reduce the used bandwidth to half the available frequency range, the signal can be resampled by 2, by retaining only every 2^{nd} point. This effectively

halves the sample rate and the spectral resolution is doubled. Care must be taken when selecting the resample rate so that aliasing is not introduced. Figure 1-2 shows that resampling is the last function performed before display processing. In time domain analysis, resampling (without introducing aliasing) has the effect of compressing the data by eliminating redundant sample point information.

IDRS also provides the capability for AM, phase, and FM demodulation. Demodulation is accomplished through a digital quadrature detector. The complex modulation envelope or base-banded signal can be represented in the form:

$$m(t) = a(t)\ e^{j\phi(t)}$$

where a(t) is the amplitude modulation term and $\sigma(t)$ is the phase modulation term. The inphase and quadrature phase terms can now be defined respectively as:

$$M_I(t) = \mathrm{Re}[m(t)] = a(t) \cos \phi(t)$$

$$M_Q(t) = \mathrm{Im}[m(t)] = a(t) \sin \phi(t)$$

The modulation signals can now be expressed as:

$$a(t) = \sqrt{M_I^2(t) + M_Q^2(t)}$$

$$d(t) = \tan^{-1} \left[\frac{M_Q(t)}{M_I(t)}\right]$$

Finally, the frequency modulation is defined as the time derivative of the phase modulation, $\phi(t)$. As part of the demodulator design, the detection filter must be designed to save only the base-band signal components. Note that control of both the predetection and postdetection filter is provided. Once demodulation is invoked, the demodulated waveforms or their spectra can be displayed for approval.

To aid in the estimation of PSD plots, IDRS provides options for waveform weighting, averaging of raw PSD's, and linear and logarithmic power scale. The options for waveform weighting are required when the PSD varies over a wide power range as a function of frequency. The application of waveform weighting serves to smooth the truncated edges of the signal segment being examined and provide a more accurate estimation of the PSD. IDRS provides three weighting algorithms (or window functions): Hanning, Hamming, and rectangular. The characteristics of these windows are discussed elsewhere.

Since real data is often processed with the system, the capability of accurately estimating the average PSD is provided. The random nature of real data suggests that any single PSD plot is not likely to be representative of the overall signal. Therefore, IDRS has the capability to average from 1 to 99 raw PSD estimates.

Finally, IDRS can display PSD information on both linear and logarithmic scales (or dB) for optimal viewing. This is a valuable viewing aid for the examination of low power spectral components.

A detailed technical explanation of the algorithms used in IDRS and software documentation is available from the Interactive Digital Receiver Simulator (IDRS) System User's Guide and Software Documentation manual #RADC-TR-7 -44. In order to provide a complete description of the IDRS capabilities, a sample waveform analysis procedure is presented in Appendix A. This sample analysis is an excerpt from the IDRS User's Manual identified above.

1.1.2. Interactive Design Philosophy

The ideal engineering aid for the investigation of waveform feature definition problems would be a mechanism through which any transformation may instantly be performed and displayed so as to highlight the discriminate traits resulting from the transformation. In the majority of problems, it will suffice to identify a reasonable subset of transformations which deal with the known scope of the problem and provide feedback to the operator in such a form that he can readily identify promising features. The ability for a system to function integrally with a user and not hinder him is a function of the design of the system structure. This concept can be termed the interactive design philosophy.

The signal processing capabilities of IDRS are shown in the block diagram of Figure 1-2. This block diagram also illustrates how the user, by setting proper logic switches, can interactively achieve the waveform-to-waveform or waveform-to-power-spectrum transformation he chooses to display. The basic architecture of IDRS is such as to allow the user to make, in the simplest and most straightforward way, requests to programs operating on the PDP 11/45 computer. These requests, which can be considered much like the function key requests of a hand calculator, are entered into the computer via the terminal keyboard by one or two character commands. These character commands set flags in the system software which act as logic switches. Figure 1-3 shows a computer-displayed menu of commands available to the user.

```
    R-RESET PARAMETERS
RL-RESET S LINES
RP-RESET S PTS/LINE
RE-RESET SKIP POINTS (RESAMPLING)
RO-RESET OVERLAP
RS-RESET SKIP POINTS

M-MU '
LO-LIST SELECTED OPTIONS

LL-LOCAL LINE SCALE
LP-LOCAL PAGE SCALE
GW-GLOBAL WAVE SCALE

W-DISPLAY WAVE
P-DISPLAY POWER SPECTRUM
PZ-DISPLAY POWER SPECTRUM-AUGMENT ZEROS
S-DISPLAY SAME DATA

X-TEXT (PAGE)
HC-RESET AUTOMATIC HARDCOPY

I-INPUT NEW TIME
T+-ADVANCE AHEAD TIME
T--ADVANCE BACK TIME

N OR NP-NEXT PAGE
WN-WAVEFORM NEXT
PN-POWER NEXT

L--PREVIOUS LINE (BACK 1 LINE)
L+-NEXT LINE (AHEAD 1 LINE)

DP-DETAILS POINTS (ITH,JTH)
DL-DETAILS LINE (EXPLODE 1 LINE)
DO-DETAILS OFF

DI-DATA ON DISK
TI-TAPE ROLLIN
TO-TAPE ROLLOUT
TD-TAPE DUMP (PAGE) FORMAT 1
TL-TAPE DUMP (PAGE) FORMAT 2
TW-WPS FORMATTED TAPE
TP-TAPE PLAYBACK (PAGE)

PA-POWER SPECTRUM AVERAGE
PL-LOG OF POWER
PP-POWER PROFILE
PW-HANNING/HAMMING/TRIANGULAR WEIGHTING

DM-DEMODULATION ON
FQ-FREQ. CONV.
DQ-PHASE/FREQUENCY
FI-FILTER INIT
WR-WAVEFORM RECTIFICATION

DC-DC REMOVAL
H ,HI-HISTOGRAM

SELECT NEXT OPTION
```

```
MO-DEMODULATION OFF
QO-FREQ. CONV. OFF
HO-WEIGHTING OFF
WO-WAVE OPTIONS OFF
PO-POWER OPTIONS OFF
FO-FILTER OFF
O-ALL OPTIONS OFF

D1-DISPLAY MODE 1
D2-DISPLAY MODE 2

FD-FILTER DESIGN
SF-SAVE FILTER PARAM.

SP-SELECT PROGRAM,SET PARAM., AND EXECUTE
EP-EXECUTE PROGRAM

K-KILL RETURN TO MONITOR
```

Figure 1-3 IDRS Menu of Options

For example, if the user wishes to take the power spectrum of the filtered raw data, he would first turn on the filter logic by giving a character command, then he would choose the power spectrum by giving another character command. The system would retrieve the raw data, skip over the band limit block (since the logic switch was not selected), then pass through the filter block to filter the data and skip over all the other transformation blocks except the FFT block to display the results.

The user may interactively design up to 10 pole Chebyshev or Butterworth digital filters. The filter can be of low-pass, high-pass, band-pass, or band-elimination type. He may also display the designed filter's characteristics such as impulse response, amplitude response, phase response, group delay, and step response for examination. The filter, once designed, remains active until it is redesigned. Of course, a new filter must be designed when starting cold on the system.

In general, the user can turn "on" or "off" any of the logic switches shown in Figure 1-2 to achieve the desired display. Once these transformed waves have been displayed, they are put into temporary storage by the system. This allows the user to do a limited amount of retransformation of a transformed wave by setting other logic switches.

The following digital processing capabilities are available in IDRS: rectification, filtering, frequency conversion, demodulation, and

derivative of phase (FM). One may display any of these processed waveforms. For any waveform or processed waveform that can be displayed on the system, the power spectrum, average power spectrum, or the log (with dB scaling) of any of them can also be displayed. DC removal, Hanning, or Hamming weighting can be performed on any waveform prior to power spectrum computations. Frequency bands can be zoomed in on as desired. The power spectrum need not be restricted to just real data. The system senses when complex power must be taken and automatically informs the user of complete system status. The system has pseudo 3-D capability for displaying power spectrums (or waveforms) plus the capability to display power profiles over time. Histogram routines are available as an aid in analysis. A one-dimensional histogram plots frequency of occurrence versus waveform amplitude. Input data may be either waveforms or power spectra.

IDRS will allow the user to interactively display on the storage tube to review, edit, or hardcopy any number of lines with any number of waveform points per line (up to 2048). The user can display selected lines and/or portions of a line on an expanded scale through a "detailed" option.

1.1.3. Data Manipulation

The waveforms are stored on a mass storage disk or mag tape and are referenced by waveform "File Name", and a beginning time (hour, min, sec). This allows the data to be randomly retrieved from the disk starting at any specified time. Once a display is up, the user can go either to the next page, previous line, next line, plus or minus X points, or select a totally new time. Those options are valid independent of the type of display.

The display ordinate, that is, waveform amplitude or spectrum power, may be scaled to local line, local page, or global wave. Scaling may be changed at any time. Text information may be added to the display. The data may be rolled from mag tape to the disk or stored from the disk to mag tape under a convenient format.

The system will allow the user to dump a page of displayed data along with headers to disk or mag tape identified by a user-entered file name. This capability allows the user to not only have a paper hardcopy of the display but also a mag tape or disk file copy. These files can be selectively played back at a later time for further manipulation of data or feature extraction.

1.1.4. Output Displays and Interpretation

The proper and speedy interpretation of waveform displays is key to successful waveform feature definition. Figure 1-4 shows a sample IDRS time waveform display page. The user is constantly informed of the options active from the list in the upper left corner of the display. As additional options are selected, they are identified in the display options list. For Figure 1-4, the following options are active:

WF	-	waveform display
LL	-	local line scale
C1	-	channel 1 displayed
FQ 6445	-	frequency shift of 6445 Hz
FL2 (CH LP 1000)	-	filter #2 active, Chebyshev low-pass at 1000 HE
RS 10	-	resample factor of 10
DEMOD DPHASE/DT	-	demodulation for AM and $\frac{d\phi}{dt}$ [FM]
DC	-	the DC component is removed
SW	-	waveforms are stored.

For up to 10-line displays, a header line is generated which indicates the beginning time for that line, effective sample rate, maximum and minimum values and the line number. For more than 10 lines, a single header line is displayed at the top indicating maximum and minimum

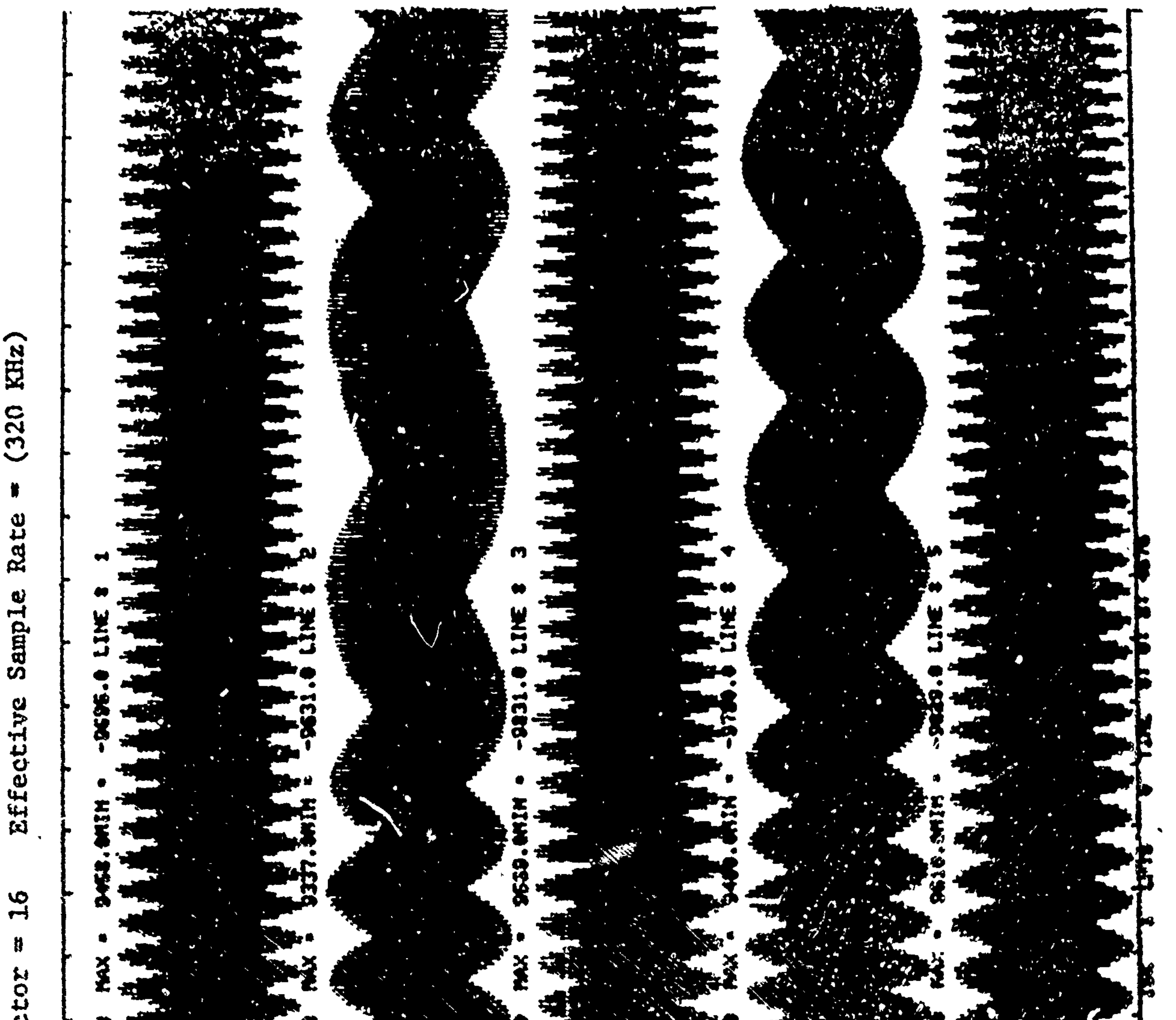


Figure 1-4 Display of a Recorded RT524 Predetection Signal

values for the page. The bottom header line is the display options list and has the following meaning by option:

ITH 1	-	first point displayed per line is #1
L/P 5	-	5 lines per page
PTS/L 1024	-	1024 points per line
ISK 1	-	every 1 point processed
LPTS Ø	-	points displayed from I^{th} point each line [detail points on if ≠ 0]
TIME	-	ending time for page HH:MM:SS:SAMPLES

Figure 1-5 shows a sample PSD plot. The current options list indicate the following:

PS	-	power spectrum plot
AVE 5	-	average of 5 raw PSD's
LP	-	local page scale
C1	-	channel 1
FQ 6445	-	frequency shift of 6445 Hz
FL2 (CH LP 1000)	-	filter #2 Chebyshev low-pass cutoff at 1000 Hz
RS 10	-	resample factor 10
HN	-	hanning weighting applied
COMPLEX	-	complex spectrum both minus and plus frequency
LOG 90 dB	-	90 dB logarithmic scale.

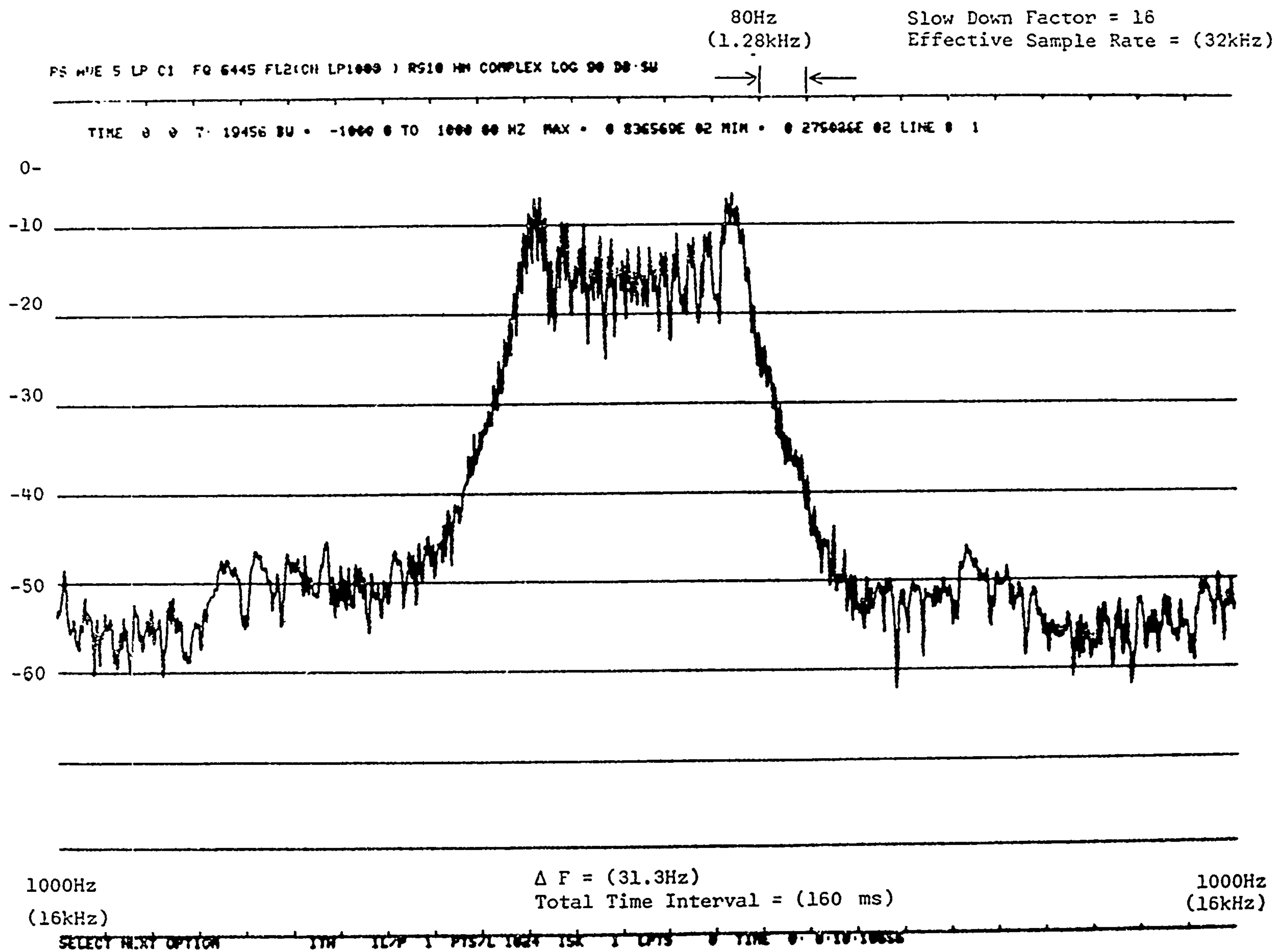


Figure 1-5 Power Spectrum of Extracted Complex Modulation Envelope After Resampling by a Factor of 10

For up to 10 lines per page, a header line will be generated which identifies the starting time for each line, frequency range which the plot displays and the maximum and minimum values for the line. The display options header line at the bottom of the page is as described for Figure 1-4.

1.1.5. Operational Environment

IDRS was implemented on a PDP 11/45 computer utilizing the RSX 11-D operating system. The graphic display terminal is a Tektronix 4014 with a Tektronix 4631 hardcopy unit triggerable manually or through software. The multiuser program development environment aids in system maintenance and the development of IDRS support software. The original implementation was capable of only single waveform storage from a fixed location on a RPØ4 disk. The IDRS software resided on a second smaller disk (RKØ5).

Since that time, the IDRS system has grown with the development of more powerful computers and operating systems. IDRS has been updated to the RSX 11-M operating system and runs on almost any PDP-11 processor with floating point hardware support. This version includes such enhancements as:

- multiuser support
- full files-11 support (multiple waveforms identified by a user assigned file name)
- inclusion of a filter characteristic storage file
- capability for utilizing IDRS for automated digital signal processing (batch processing capabilities)
- waveform manipulation via standard system utilities.

IDRS is also running under the VMS operating system on the DEC VAX-11/780 computer series. A high speed version of IDRS running under RSX 11-M utilizes a DEC PDP 11/FPS AP120 Array Processor parallel processing scheme.

1.2. IDRS AS A PATTERN RECOGNITION TOOL

Past experience suggests that for effective pattern analysis of waveform data as a prelude to waveform feature extraction, an easy-to-use flexible interactive computer system with sufficient graphics to exploit the operator's ability to recognize data structures is required. The capability for the operator to effect and analyze complex data transformations quickly permits the structure analysis of many waveforms for an accurate representation of the data base.

The functionality of IDRS as a pattern recognition tool lies in its application as a waveform structure analysis and transformation capability. As discussed in Section 1, IDRS is a work bench of receiver modules interfaced with powerful spectral analysis and display capabilities. The wide variety and flexibility of the modules permits the application of IDRS to the general realm of signal processing problems. IDRS is especially powerful, however, in the area of emitter structure analysis. Here, all a priori information (and hypotheses) can be incorporated into a receiver designed specifically to investigate suspected discriminate traits.

1.2.1. Waveform Structure Analysis/Transformation

Perhaps the single most desirable quality of a waveform feature definition system is its ability to quickly effect a wide variety of complex data transformations so that the operator can identify discriminate traits which become apparent only in a transformation space. The time domain analysis of raw waveform data provides only the most superficial characteristics of data. Often, these characteristics, (amplitude, periodicity, and nature of envelope, for example) are enough. More complicated waveform recognition problems will probably require additional characteristics not readily observable from the raw data time domain analysis. Although no single transformation algorithm is the answer to all waveform recognition problems, the variety of transformations available in IDRS are applicable to the majority of waveform feature definition problems.

1.2.1.1. Digital Filtering

The IDRS provides a flexible digital filter capabilitv where the user specifies the filter on-line. The IDRS then designs and realizes the filter. The user may then ask to see displays of the performance characteristics of the resulting filter (see Figure 1-6 for a sample of the displayed filter characteristics). If the user is satisfied with the performance, he will command IDRS to implement the filter in an over-all processing sequence. If the user is not satisfied with the filter performance, he can alter his specifications, and the design process is repeated.

The digital filter design choices presently in IDRS are recursive filters, either a Chebyshev or a Butterworth. The user specifies whether he wants a low-pass, high-pass, band-pass, or band-reject response. He also specifies the cut-off frequencies and number of poles (up to 10). The pass-band ripple is presently fixed at 1 dB.

Digital filtering provides a means by which the operator can smooth waveforms, suppress dominant features to enhance low power signal components, extract frequency bands, etc. Through the ability to control both the filter cutoff and the roll-off characteristics, the user has much power to isolate or shape fre ency bands of interest to further analysis.

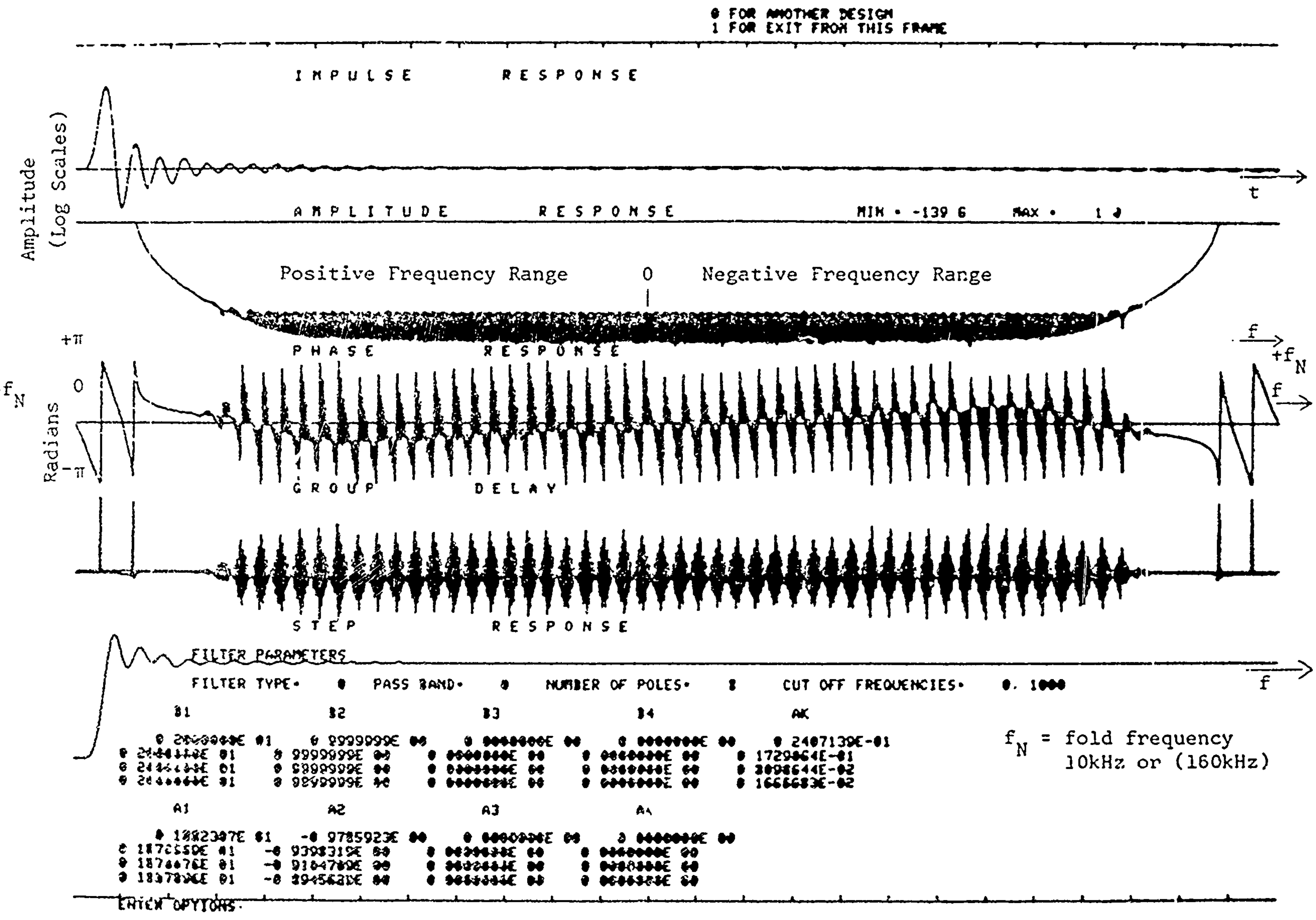


Figure 1-6 Display of Filter Characteristics

1.2.1.2. Demodulation

The IDRS extracts amplitude modulation (AM), phase modulation (PM) and frequency modulation (FM) waveforms from the quadrature waveforms:

$$a(t) = m_I^2(t) + m_Q^2(t)$$

$$\phi(t) = \tan^{-1} \frac{m_Q(t)}{m_I(t)}$$

$$\phi(t) = \frac{\phi(t) - \phi(t - \Delta t)}{\Delta t}$$

where

t = time index

Δt = sampling interval.

The detailed discussion of the demodulation techniques is available elsewhere.

The ability to extract the AM, FM, and Phase component waveforms and their spectra often leads to a new basis for discriminate analysis. For example, analysis of waveform components and their spectra for data transmission can provide information of based rates, phase reversal transients, local oscillator stability, etc. Of course, it is possible to first band-limit the signal to a particular frequency band through digital filtering, then apply demodulation.

1.2.1.3. Fast Fourier Transform (FFT)

IDRS provides the capability to view any point in the receiver sequence in both the time and frequency domains. Associated with the transformation to the frequency domain via the FFT are a series of options which tailor the power spectral density (PSD) estimation. The application of a weighting function can provide a better estimation of the PSD.

Since many of the signals to be analyzed are made up of random components, at least in part, any single or raw power spectrum estimate (periodogram) is likely to be a poor representative of the average properties of the signal. Therefore, it was necessary to give the IDRS user the option to average a number (1 to 99) of raw PSD estimates or periodograms.

The third user option needed for power spectral density analysis was that of selection of scale in displaying PSD plots. In addition to a linear power scale a logarithmic (or dB) scale is required because of the wide dynamic range of power spectra associated with signals to be analyzed. Thus, the IDRS user can select either a linear or logarithmic scale for displaying PSD plots.

The fast Fourier transform used in IDRS is limited to a maximum of 1024 points. The resolution provided by this may sometimes not be enough

to display the fine grain structure of the spectrum. One way to increase the resolution is to increase the FFT capability to more than 1024 points. However, this upper limit of 1024 points has been dictated by the memory space available in computer hardware. An alternate scheme is implemented in IDRS that increases the resolution to any order. The block diagram of this scheme is shown in Figure 1-7. First, the band of frequency around which the magnified resolution is to be obtained is translated to zero frequency. Then a low-pass filter is used to curtail the bandwidth of the signal to a value for which the magnified resolution is required. Now, since the bandwidth of the low-pass signal has been reduced, it can be resampled. A 1024 point FFT of this resampled wave will provide an increased resolution of the desired spectrum band. Of course, this scheme assumes that more than 1024 points are available in the original waveform so that it can be resampled to obtain 1024 points.

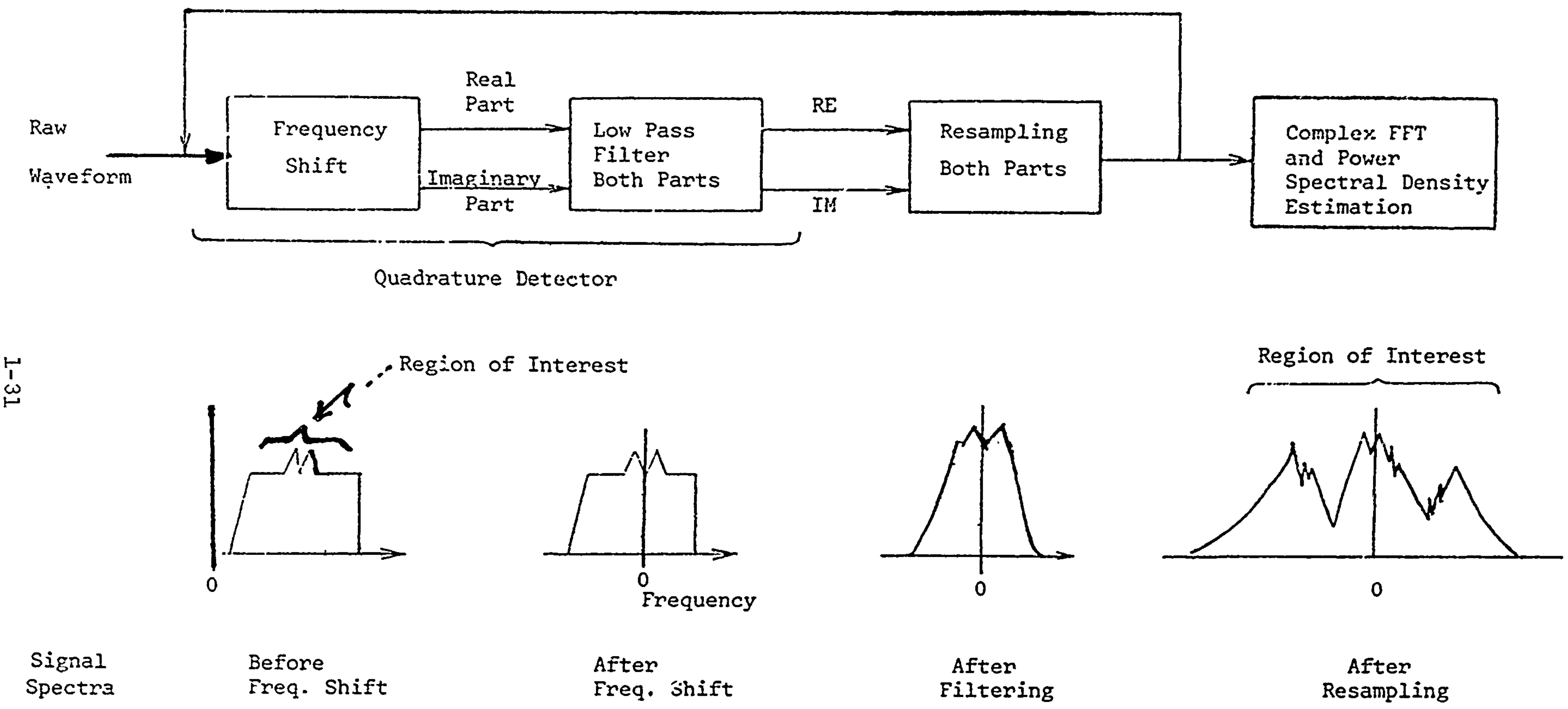


Figure 1-7 Block-Diagram of Power Spectrum: Zoom-In Scheme

1.2.1.4. Waveform Histograms

An additional method for extracting information from raw data is the use of a histogram. We speculate that possible features might, for example, be indicated by the symmetry of a histogram, or by some special trait of the tails of a histogram.

The IDRS histogram option displays frequency of occurrence vs. waveform amplitude. IDRS can generate a histogram of any waveform or power spectrum. Figure 1-8 shows a histogram of a waveform which oscillates between two values.

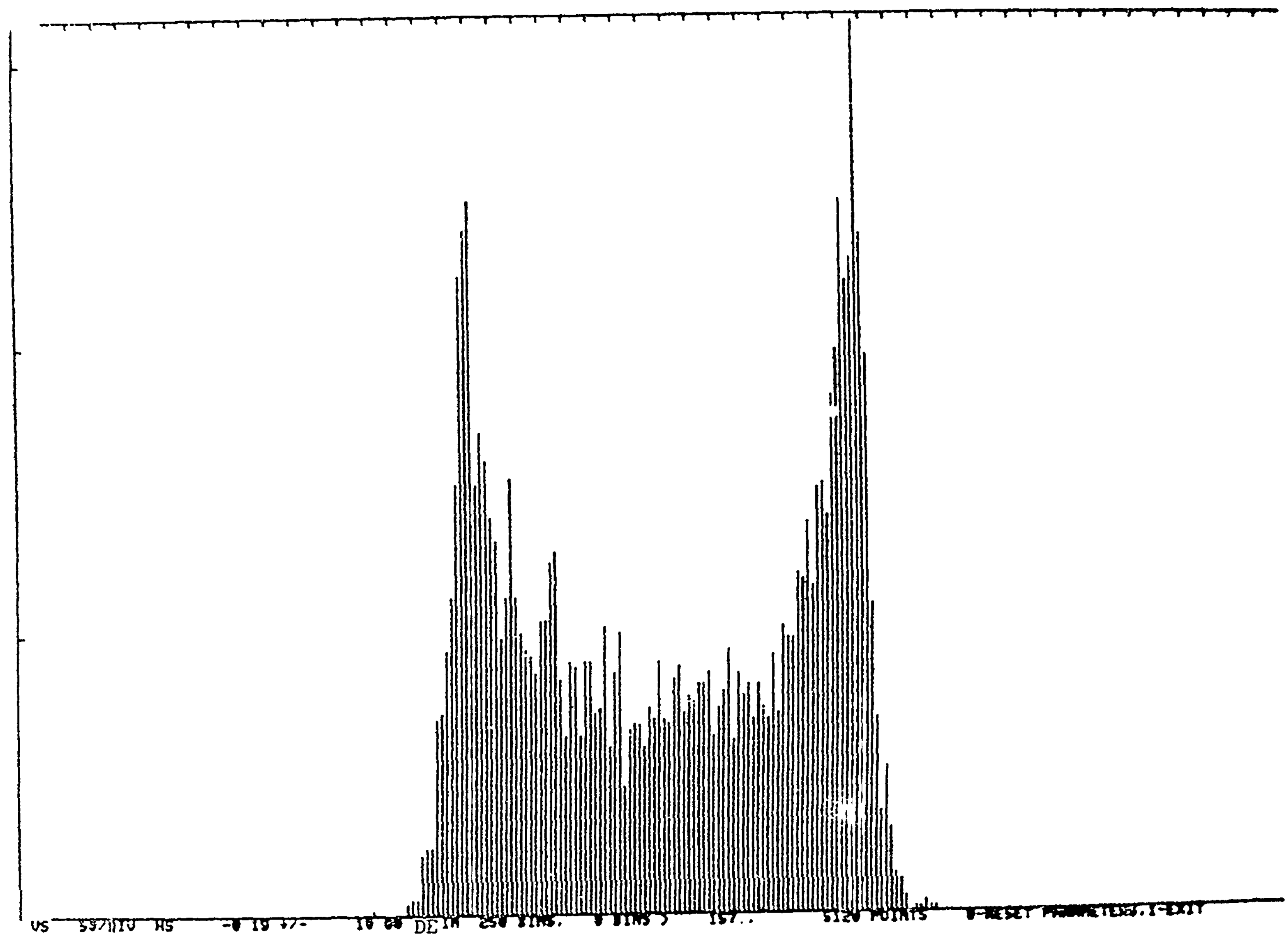


Figure 1-8 Histogram of FM Waveform

1.2.2. Feature Extraction

Once a promising transformation has been determined, it is necessary to extract key measurements from the transformed data. This operation is accomplished external to IDRS. The IDRS is a feature definition tool. Once the features have been identified through appropriate transformations, IDRS provides the capability of writing out a display page to disk or mag tape.

Once a sequence of IDRS options has been identified to provide the required transformation, IDRS can be run in a batch mode to perform this same sequence of options on all of the waveform(s), saving the display pages on disk or mag tape, identified by file name. This process is analogous to performing the identified transformation on a list of waveform files to produce new transformed versions of the same waveforms.

The complexity of a measurement extraction capability which would meet all possible requirements, if such a capability could be devised, would be extreme indeed. An attractive alternative, however, is the development of a specialized measurement extraction module (in Fortran) which performs only that operation required and optionally formats the measurement set into OLPARS vector format.

Since the waveform file outputted from IDRS is in a standard format, it is straightforward operation to read in a waveform and extract a set ot feature vectors. This is a capability of the PDP 11 RSX 11-M and VAX VMS operating systems Command File Processing Languages. It provides an easy mechanism for a waveform tree to be sequentially processed through IDRS, and for a user-written measurement extraction task to generate a tree of feature vectors for OLPARS. analysis.

BIBLIOGRAPHY (Section 1)

1. Proctor, A.H.; White, D.D., "The Long Waveform System," Rome Air Development Center Report to be published.

2. Bozek, R.E.; Holman, R.H., et al, "Interactive Digital Receiver Simulator (IDRS) System User's Manual and Software Documentation," RADC-TR-78-44, March 1978. (A053684)

3. Oppenheim and Schafer, Digital Signal Processing, Prentice-Hall, Inc., New Jersey, 1975.

4. Gold and Rader, Digital Processing of Signals, McGraw-Hill Book Company, New York, 1966.

5. Natali, F.D.; Magil, D.T., "Digital Processing Receiver Study," Technical Report No. RADC-TR-68-163, May 1968, AD #835767.

6. Fritchman, B.; Gumacos, C.; et al, "Digital Equivalent Transceivers Study," Final Technical Report RADC-TR-68-539, March 1969, AD #851364.

APPENDIX A

For Section 1

The following is an example of a sample feature investigation using IDRS. The example originally appeared in the document: "The Interactive Digital Receiver Simulator (IDRS) User's Manual and Program Documentation ," RADC TR-78-44, March 1978. (A053684)

1.4. SAMPLE RESULTS OF DIGITAL PROCESSING OF A PREDETECTION SIGNAL

This section takes the reader through a typical IDRS processing sequence. Hardcopy displays for each step are presented and discussed.

The discussion begins with a sample of a digitized signal stored on the computer system disk. The signal is a predetection recording of an RT-524 transmission, that was originally analog recorded at a 100 kHz carrier frequency. The analog tape recording was played-back at 1/16th of the recording speed and digitized at a rate of 20K samples/second. Thus, the effective sampling rate was 320K samples/second.

1.4.1. Plot of Predetection Waveform

The user requests a display of a portion of the stored digital signal. Figure 1-9 shows a steady-state portion of the RT-524 predetection recorded waveform.

The computer-produced annotation at the top of Figure 1-9 indicates to the user the processing operations that have been selected to produce the displayed result. "WP" indicates that a waveform is plotted. "LP" indicates that the waveform amplitude scaling is local page, that is the scaling is set based on the maximum waveform value appearing on the page. The user also has the option for local line (LL) scaling where each line of a display is scaled according to the maximum waveform value contained within the line of data. "C1" indicates that the data was retrieved from the data disk Channel 1 file. "REAL" indicates that the real part of the waveform is plotted (in this example, the waveform itself is real).

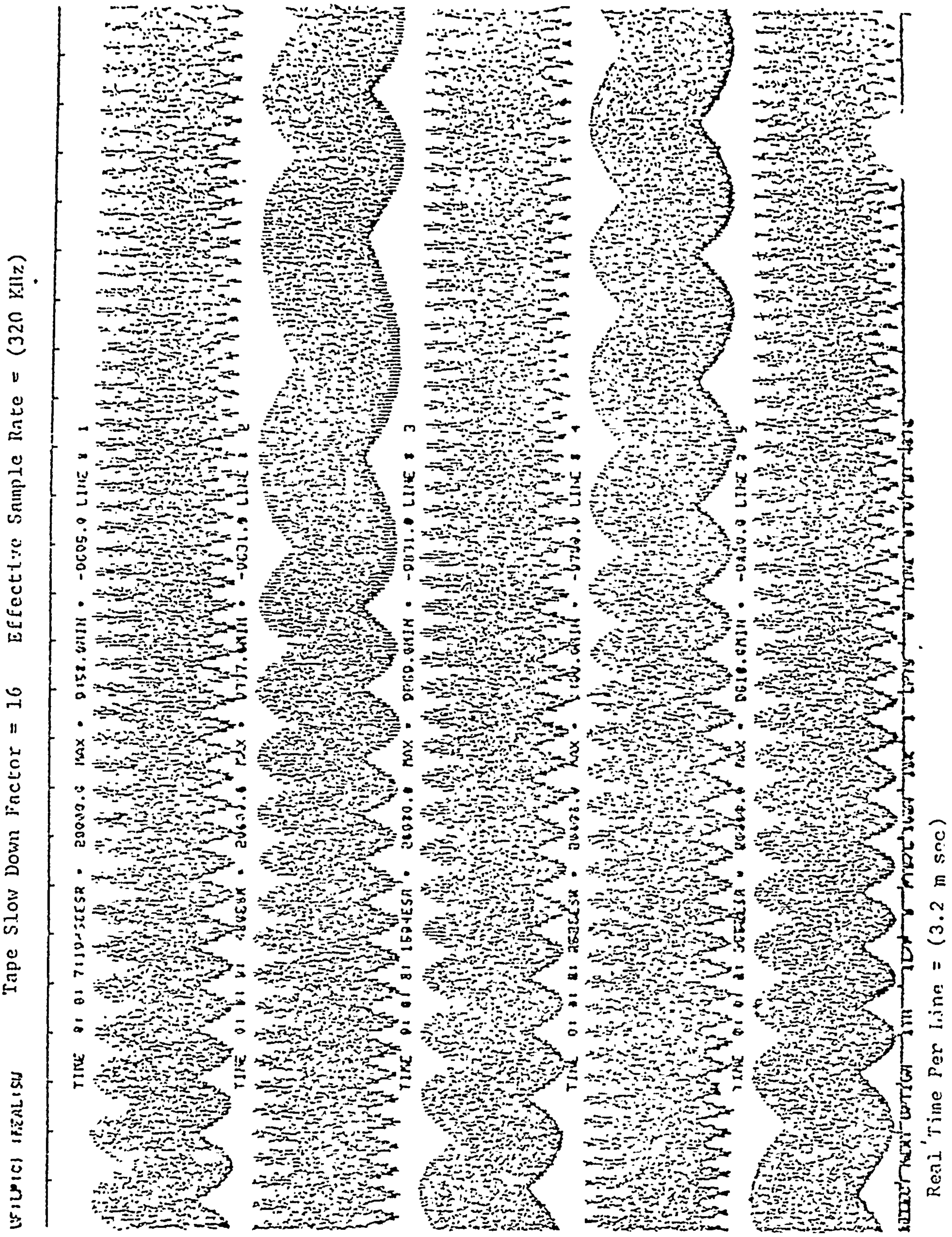


Figure 1-9 Display of a Recorded RT524 Predetection Signal

Referring to the computer-produced anotation at the bottom of the display, "SELECT NEXT OPTION" indicates to the interactive ' 'er that the system is ready for his next command. The "Ith" along with . "Jth" option allow the operator to select a waveform plot starting and stopping points for plotting a portion of a line of data with increased time scale. In Figure 1-9 the user elected to plot each entire line of data (e.g. I=1). "L/P 5" indicates that the user has elected to have displayed 5 lines of data per page. The user has the option of having from 1 to 99 lines displayed per page. "PTS/L 1024" indicates that the user has elected to have 1024 data points plotted per line. The user has the option of selecting from 1 to 2048 data points per line. 1024 plotted data points per line were selected to correspond to the 1024 points maximum size FFT that IDRS can calculate. "ISK 1" indicates that the user has selected to have no skip points and thus to have successive points plotted. If the user had selected ISK equal to 2, every other point would have been plotted. If ISK had been selected equal to 3, every third data point would have been plotted and so forth. A maximum ISK of 99 can be selected. "LPTS 0" indicates that the user selected to have zero overlap with each of the data lines 2 through 5 with respective to its previous line. The user may select to have an overlap of from 0 to the selected value of "PTS/L". "TIME 0:0:8:4576" indicates the starting time of the next line of data if it were displayed.

Computer-produced annotation also appears for each line of data in Figure 1-9. Using, for illustration, the annotation associated with and appearing above the first line of data, a starting time is given for that line of data: "TIME 0:0:7:1945". The format for the time annotation is,

Hours : Minutes : Seconds : Fraction of a Second (expressed in a number of sample interval counts).

It is important to point out that the indicated time refers to digitizing time and not signal time. Signal time is obtained by dividing ADC time by 16, the analog tape slow down factor. Next, "ESR = 20000" indicates that sample rate is known to be 20K samples per second. This was the sample rate of the analog-to-digital converter (ADC). Also, the maximum and minimum values are given for each line in units of ADC levels.

When more than 10 lines of data per page are requested, that "line" annotation is given for only the first line. The start time for the first line indicates the start time for the page.

The user accesses data by specifying the start time. The user advances from one page to the next by typing in "N" for next. The effective data sample rate is 320K samples/sec., and thus each line of data contains 3.2 msecs. of data referenced to real time. The parentheses around "quantities on display" annotation in figures in this section refer to real time as opposed to ADC time.

It is observed that the predetection digital signal shown in Figure 1-9 exhibits an envelope structure. This envelope structure is not symmetrical about the mean signal amplitude as would be the case for true amplitude modulation. The observed envelope is a beat relationship between the frequency of the digital signal and the sampling frequency. This occurs even though the sampling rate has been selected to meet the Nyquist sampling criterion as illustrated in Figure 1-10. The apparent amplitude modulation demonstrates an important factor. The point is that in order to obtain a plot of waveform samples that appear highly similar to the original analog waveform, the waveform must be sufficiently oversampled. Thus, oversampling of waveforms is desired for human analysis and interpretation. Even though the waveform plotted in Figure 2-9 is not sufficiently oversampled, an oversampling condition occurs as a result of subsequent digital signal processing.

Inspection of the waveform displayed in Figure 1-9 reveals a periodic structure with a period equal to approximately two lines of data or frequency of 156Hz. This periodicity is caused by a squelch side-tone FM modulation of the RT-524 signal, nominally at a 150Hz rate of deviation.

A display of the predetection waveform, as in Figure 1-9, serves as a coarse grain preview of the signal in time domain. A time domain preview, along with a frequency domain preview, allows the user to find a signal portion suitable for further processing. For example, if the user desires to develop a spectral plot requiring "n" lines of data to be averaged, he would want to make a coarse check to be certain that the signal was of sufficient duration to support an averaging of "n" lines of data. He may also wish to observe a coarse signal spectrum plot versus time.

1.4.2. Two-Dimensional Plot of a Power Spectral Density

Figure 1-11 shows an IDRS hardcopy plot of the power spectral density PSD of the signal displayed in Figure 1-9. The computer-generated annotation, "PS", at the top of the display indicates that the user has selected the power spectrum processing option. Furthermore, the user has asked for an average of 20 raw spectra, as indicated by "AVE:20". The annotation "LOG" indicates the selection of logarithmic power scale. Ten dB lines, referenced to the spectrum peak, are generated on the display to aid the user in his interpretation. It is noted that the computer-produced annotation indicates the frequency scale to be from 0 to 10KHz (the fold frequency). This frequency scale refers to ADC time. The signal frequency scale is obtained by multiplying the ADC frequency scale by a factor of 16. Thus, the signal frequency scale covers from 0 to 160KHz.

Each of the raw spectra making up the average PSD plot results from an FFT of a line of data containing 1024 time samples. Since a real signal is Fourier transformed, a power spectrum results that is symmetrical about the frequency origin. For this reason only, the positive half of the PSD is plotted, resulting in 512 frequency points per line. The resulting real time

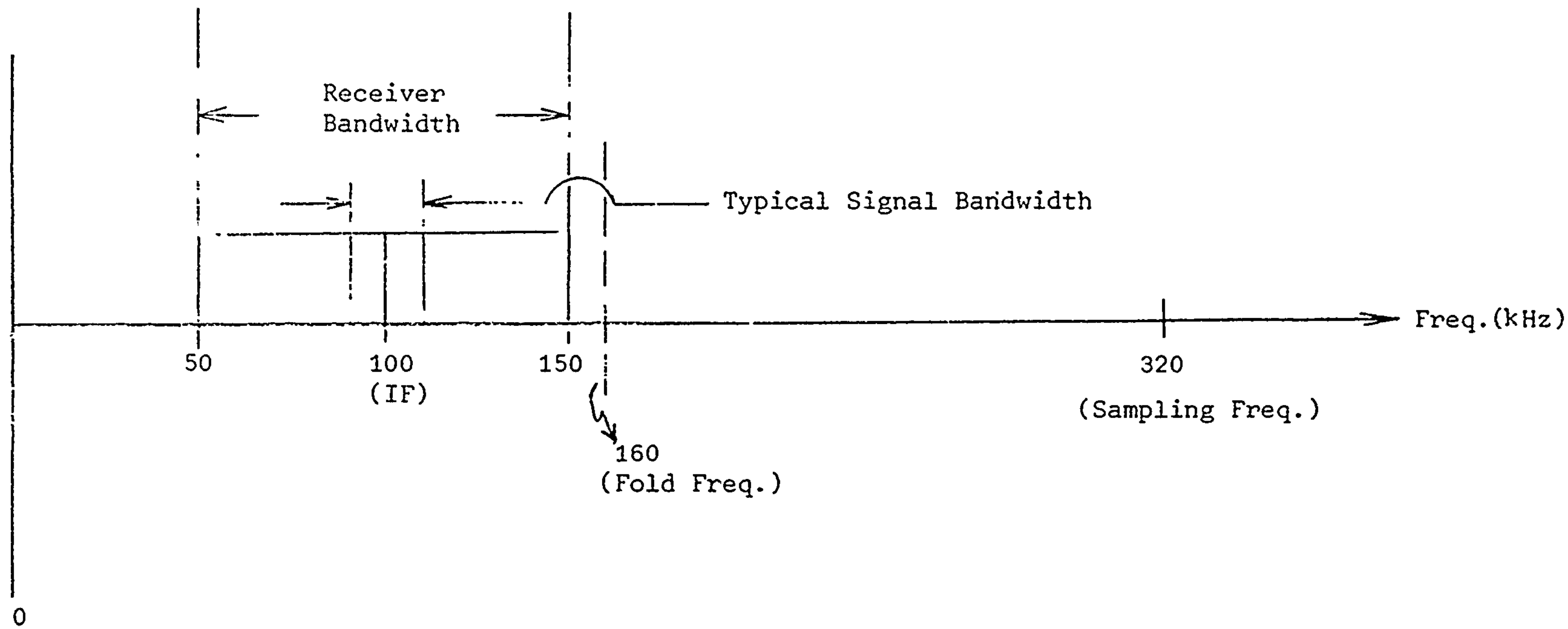


Figure 1-10 Relationship of Recorded IF Band and Sampling Rate

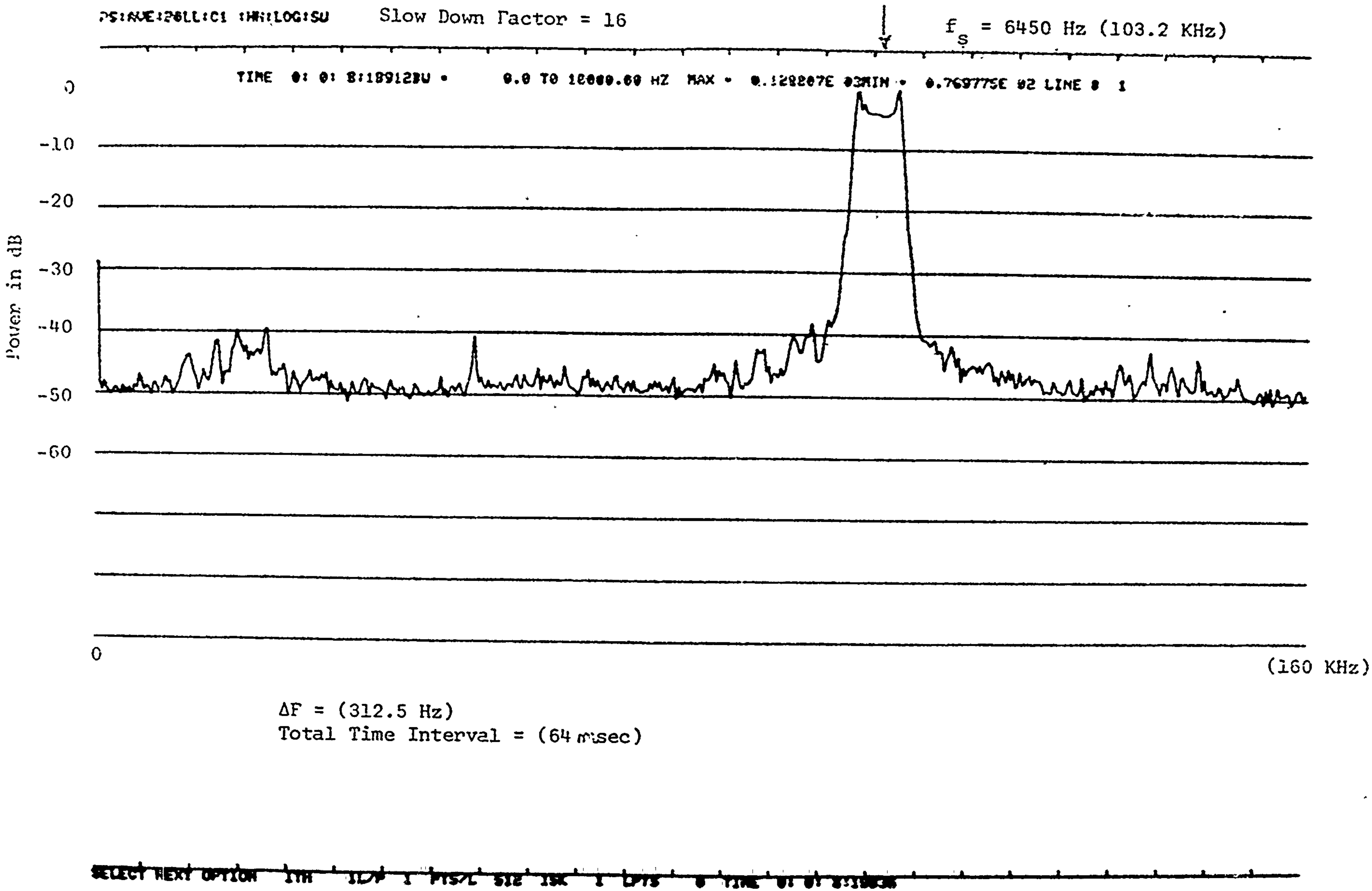


Figure 1-11 Display of Power Spectrum of a RT524 Predetection Signal With Hanning

frequency sampling interval, Δ F, is equal to the real time sampling fold frequency, 160KHz divided by 512, that is $\Delta F = 312.5Hz$.

The remainder of the computer-produced annotation has been defined previously.

The shape of the observed signal spectrum results from the RT-524 side-tone FM. Because this is wideband FM, the width of the spectrum is approximately equal to the peak-to-peak frequency deviation. This width is observed to be approximately 6KHz. The individual spectral lines, due to the approximately 150Hz sinewave modulation, are not resolved in the plot. The spectrum shape is, however, that of a classical wideband sinusoidal FM modulation. It is noted that sidebands other than due to the side-tone modulation appear to be 40 dB or more below the signal.

Since the actual carrier is not resolved, it has been estimated by the user to be the center of the signal spectrum. That is 6450 Hz (103.2 kHz in signal time). This number was then used as the shift frequency in the subsequent quadrature detection.

Also, the spectrum shown in Figure 1-11 was obtained from the waveform after a Hanning weighting had been accomplished. The annotation "HN" on the display indicates that the Hanning weighting had been applied. Figure 1-12 shows the spectrum of the same data without Hanning weighting. The advantage of the Hanning weighting is clearly evident in that additional detail of the spectrum is obtained at frequencies near the principal spectral lobe.

1.4.3. Plot of Power Spectral Density as a Function of Time

Figure 1-13 shows a display of the predetection power spectrum versus time on a linear power scale for the signal previously displayed at the same starting time. The user has selected 99 lines per page (L/P) and a line overlap (LPTS) of 900 samples. The selection of these parameters causes the display to have a three-dimensional appearance where relative power is observed as the height of the image and as a function of the frequency and time.

The parameters selected for the power versus frequency display shown in Figure 1-13 are such as to reveal nicely the squelch FM. The instantaneous frequency is observed to vary periodically. In general the power spectrum versus time plot is highly useful for displaying signal trends.

1.4.4. Plot of Signal Spectrum after Frequency Translation

Figure 1-14 shows the computer-generated PSD plot of the same signal sample but after frequency translation. Since a complex signal results from the frequency translation, both the negative and the positive halves of the calculated spectrum need to be displayed. Thus, in Figure 1-14, zero frequency appears in the center of the display and the shifted positive

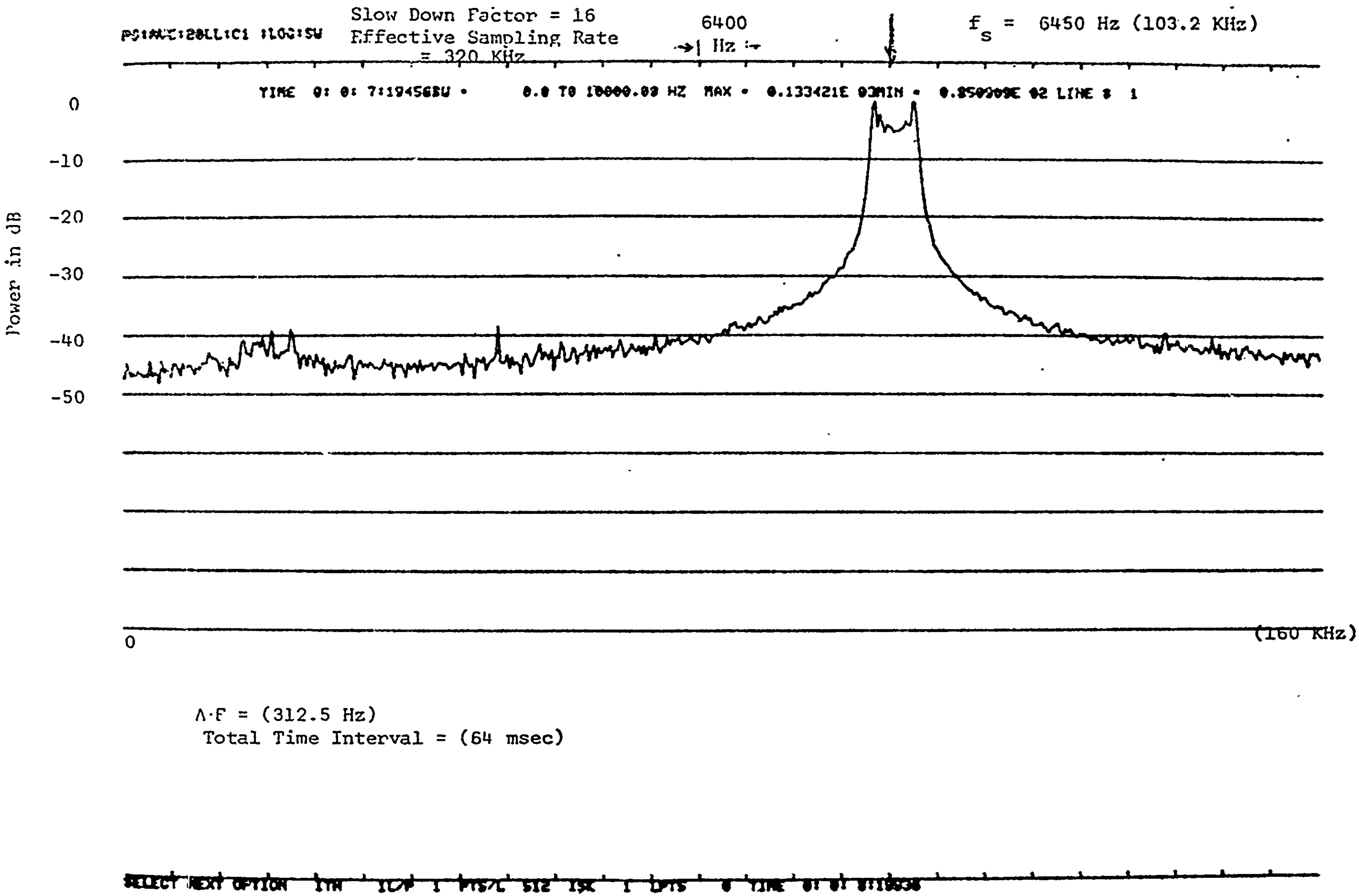


Figure 1-12 Display of Power Spectrum of a RT524 Predetection Signal Without Hanning

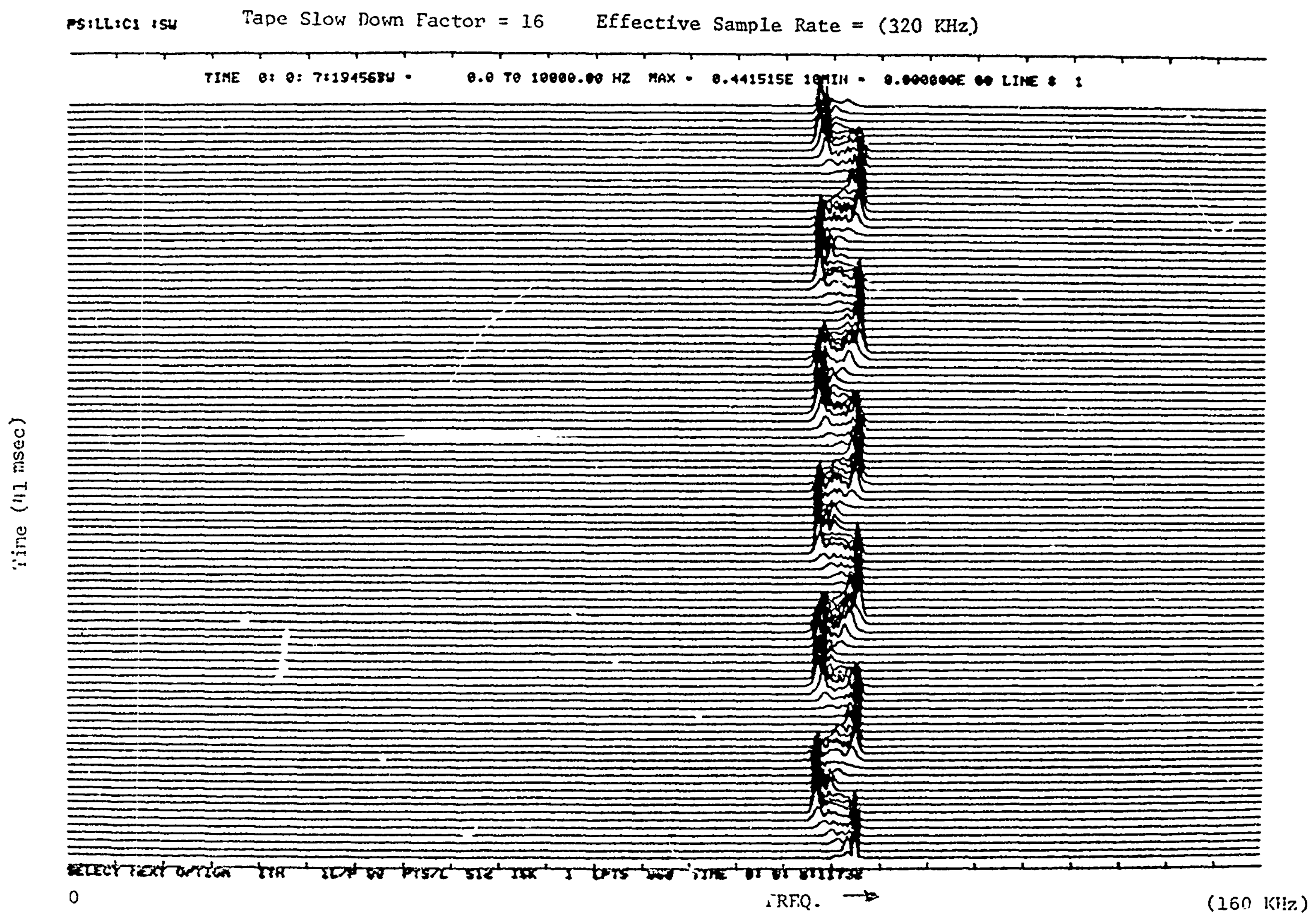


Figure 1-13 A Power Spectrum (One Linear Scale Per Line) vs. Time of a RT524 Predetection Signal

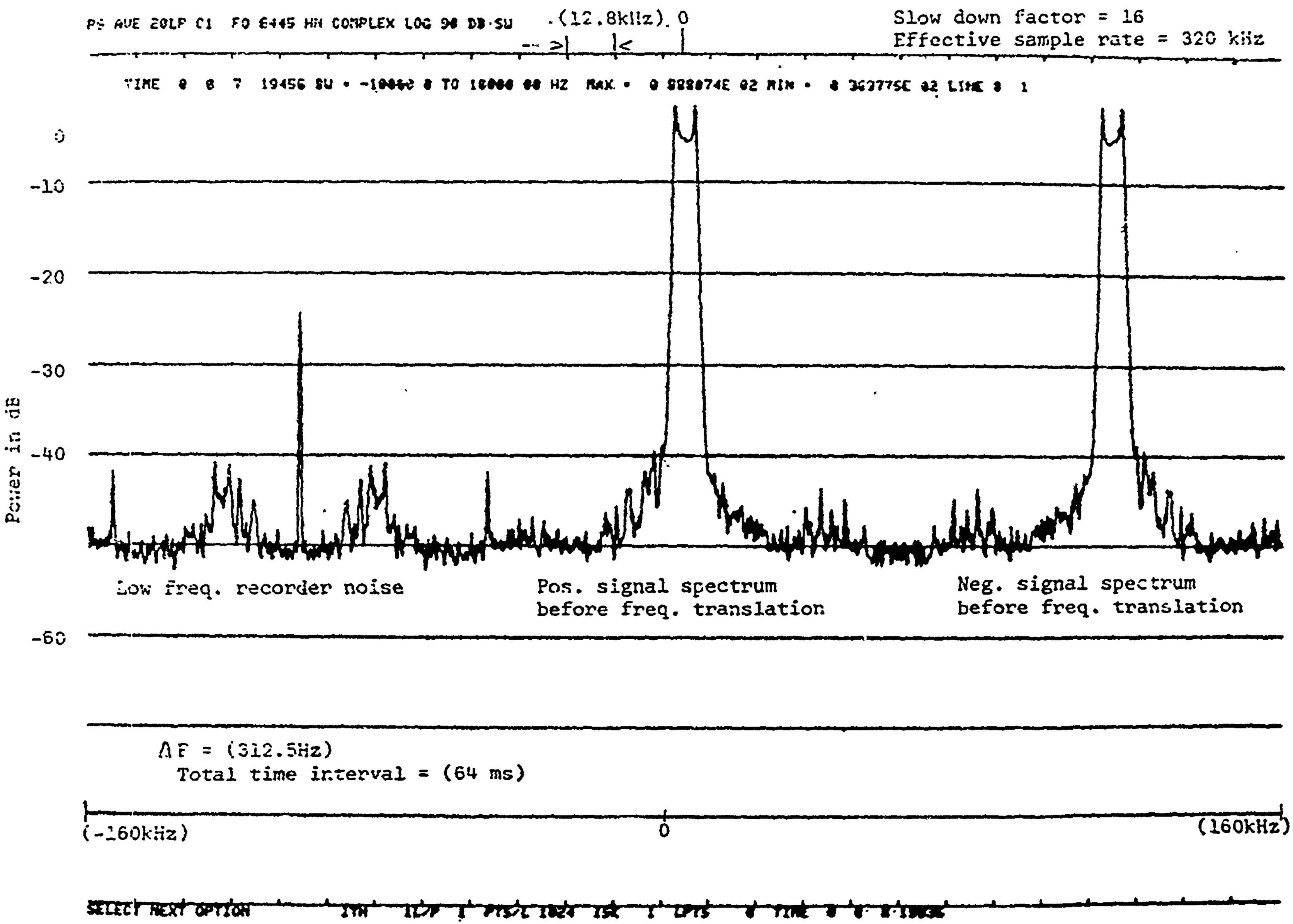


Figure 1-14 Power Spectrum of Signal After Frequency Shift

frequency portion is now centered at approximately zero frequency. The negative frequency portion of the signal spectrum is now shifted so that it appears in the positive frequency range below the positive fold frequency (160 KHz). The spectrum of the low frequency additive noise associated with the analog recorder is now shifted so as to appear in the negative frequency range. Except for the frequency shift and display of both the negative and positive frequency ranges, the PSD plot in Figure 1-14 was obtained with the same processing parameters as the PSD plot in Figure 1-11. It is noted that the frequency shift is annotated on the spectrum plot by "FQ 6445."

1.4.5. Waveform Plot of Frequency Translated Signal

Figure 1-15 shows a computer-generated waveform plot of the real part of the complex signal that results from the frequency translation. The five lines of data shown correspond precisely to the original sample shown in Figure 1-9 and the data starting time is the same as used in developing the above-discussed spectral plots.

Inspection of the waveform reveals high-frequency signal superimposed on a low-frequency signal. The high-frequency signal corresponds to the high-frequency spectrum, and the low-frequency waveform corresponds to the base-banded spectrum shown in Figure 1-14.

The effect of the squelch side-tone modulation is clearly evident in the low-frequency component of the waveform. Since the low-frequency spectrum is approximately centered at zero frequency, the FM side-tone modulation will cause the instantaneous frequency of the complex signal to alternate between positive and negative frequencies at the side-tone modulation rate. Thus, the signal goes through a time of "zero" frequency twice in each period of the modulation rate. These transitions through zero frequency are clearly evident in the real waveform plot, occurring once on each line of data.

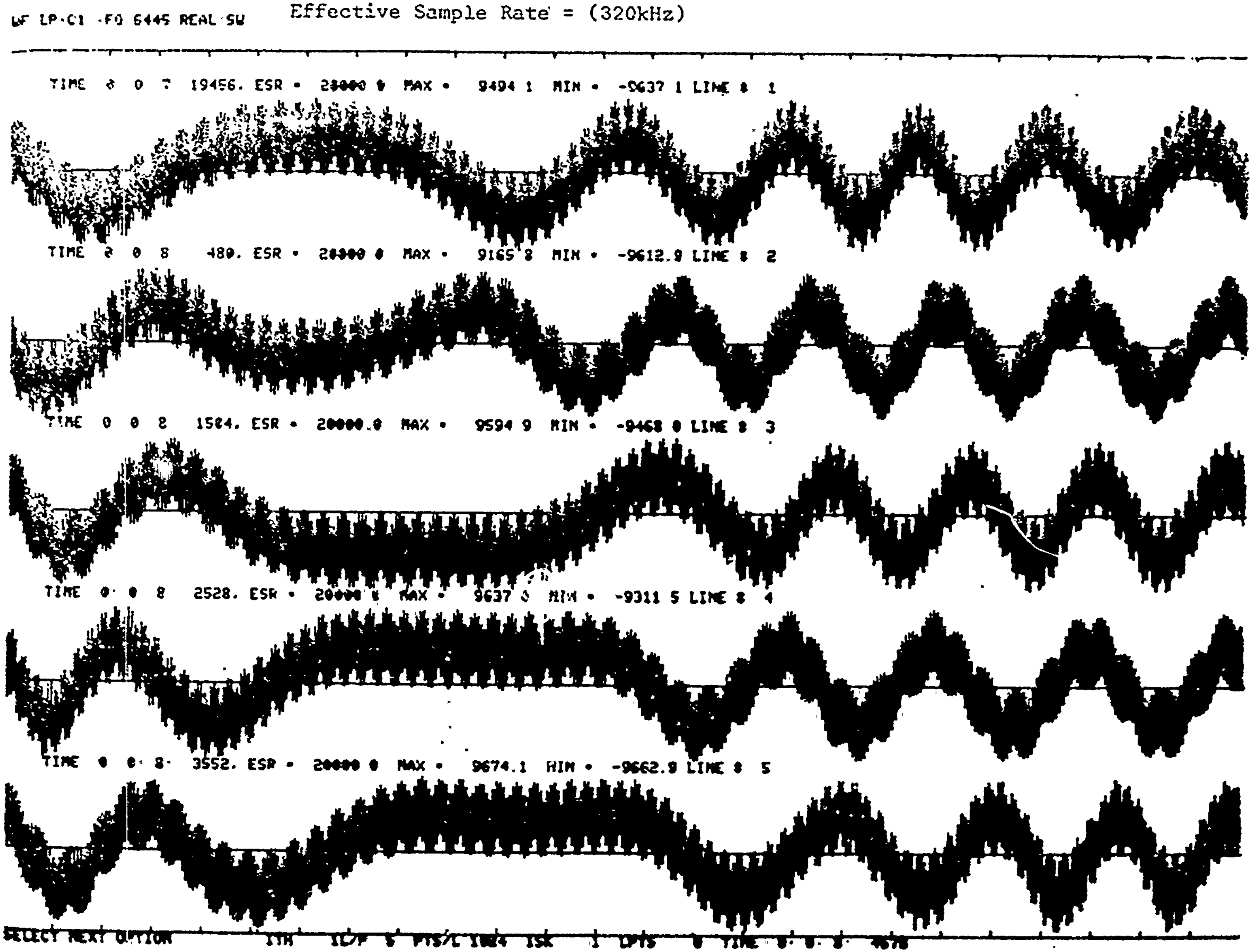
The imaginary part of the complex waveform, if plotted, would appear quite similar to the real part.

1.4.6. Plot of Signal Spectrum after Frequency Translation and Filtering

The next step in the processing scheme was to filter the complex waveform that resulted from the frequency translation. The filtering accomplishes two things. First, the high frequency term is rejected so that the signal complex modulation envelope results in terms of in-phase [$m_I(t)$] and quadrature-phase [$m_Q(t)$] waveforms. Secondly, the bandwidth of the filter can be selected so that the filter serves as a predetection filter for rejecting unwanted noise and interference.

Figure 1-16 shows the computer-displayed characteristics of the designed filter. The annotation "FILTER TYPE=0" indicates that a Chebyshev filter was selected by the user; "PASS BAND=0" indicates that a low-pass filter form was

Slow Down Factor = 16
Effective Sample Rate = (320kHz)



(Real Time/Line = (3.2ms)

Figure 1-15 Waveform Plot of Real Part of Frequency Translated Waveform

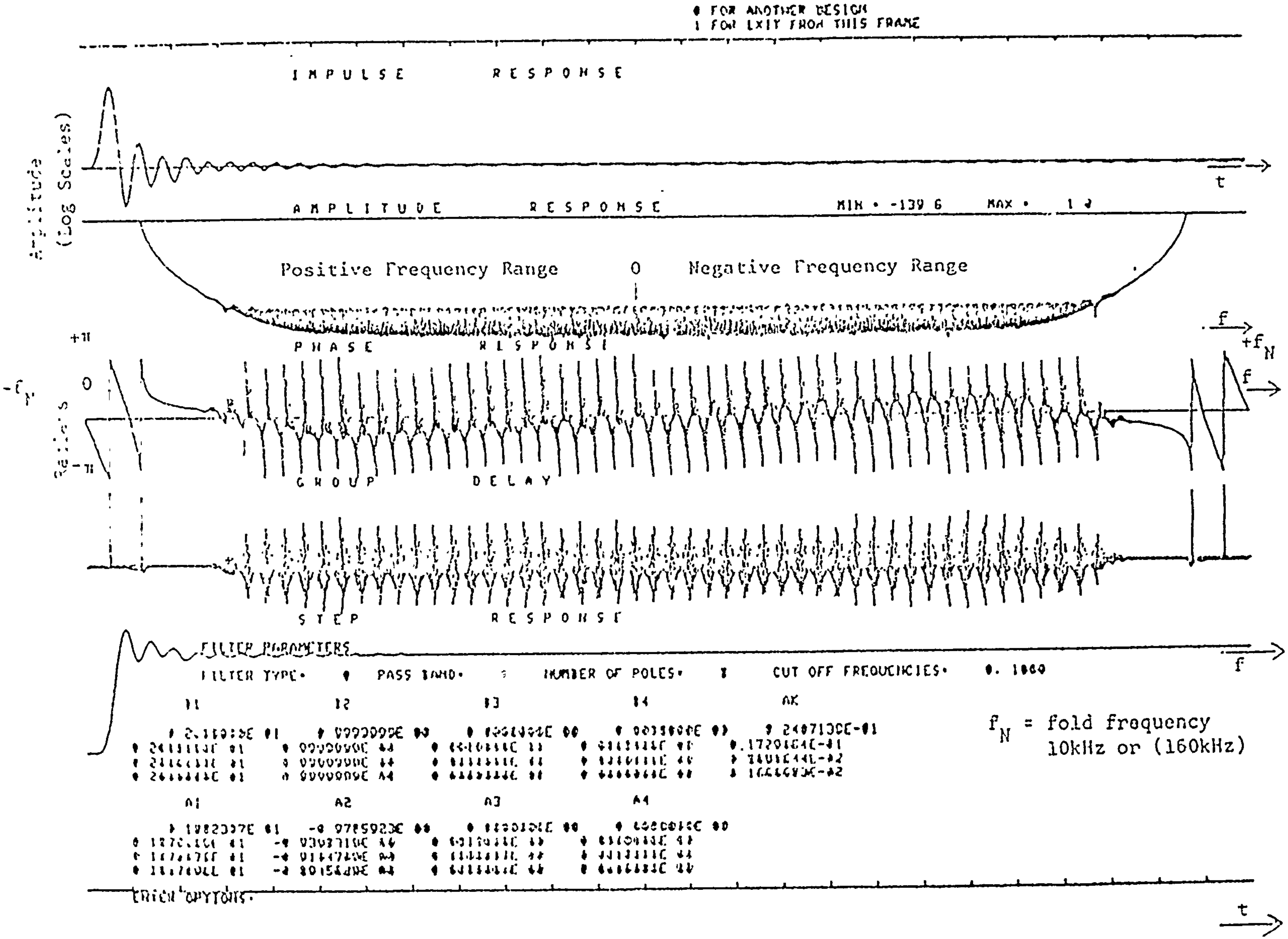


Figure 1-16 Display of Filter Characteristics

selected; "NUMBER OF POLES=8" indicates that an 8 pole filter design was elected; and "1000" indicates the selected filter cutoff frequency, referred to ADC - time. The cutoff frequency referred to real time is 16 kHz.

The top plot in Figure 1-16 shows the time response of an impulse applied at the beginning of a line of data. The fifth plot likewise shows the time response due to a step input. The second plot shows the filter amplitude response on a log scale versus frequency. The annotations "MAX=1.0 MIN=140.0" indicate a 1 db ripple in the passband and a maximum attenuation of 140 db in the stopband. The third plot likewise shows the phase versus frequency response. The fourth plot shows the group delay as a function of frequency.

The phase-versus-frequency plot shows that the filter in-band phase response is not perfectly linear. The folds in the in-band phase response are due to the RSX FORTRAN arc-tangent routine and thus are not associated with the filter. Since the group delay is the derivative of the phase response, the folds cause the in-band spikes in the group delay plot. These also are not indicative of the filter.

The impulse and step responses appear to be more useful to a user in observing signal time delay through the filter.

Figure 1-17 shows the plot of the PSD of the frequency-translated signal after digital filtering. The filter was identically applied to the real and imaginary waveforms. The annotation "FL2(CH LP 1000)" at the top of the plot, specifies that a Chebyshev filter with a low-pass cutoff frequency of 1000 Hz was used. The annotation "COMPLEX" indicates that the spectrum plot resulted from a complex waveform.

The predetection filter bandwidth was 32kHz. The effectiveness of the rejection is evident from observing the plot.

1.4.7. $m_I(t)$ and $m_Q(t)$ Waveform Plots

Plots of the $m_I(t)$ and $m_Q(t)$ waveforms are shown respectively in Figures 1-18 and 1-19. It is evident from comparing the plots to figures that only information relating to the low-frequency spectrum is retained.

Transients are evident in the waveforms at the beginning of the first line of data in both plots. The transients are due to the fact that the digital filtering process is initiated at the indicated page-starting time with the digital filter in a "zero" state.

It is noted that these waveforms are now well over-sampled, that is, the appearance of plot will not change if the sampling rate is reduced by a factor 2 or 3. This over-sampled condition is desired for visual analysis.

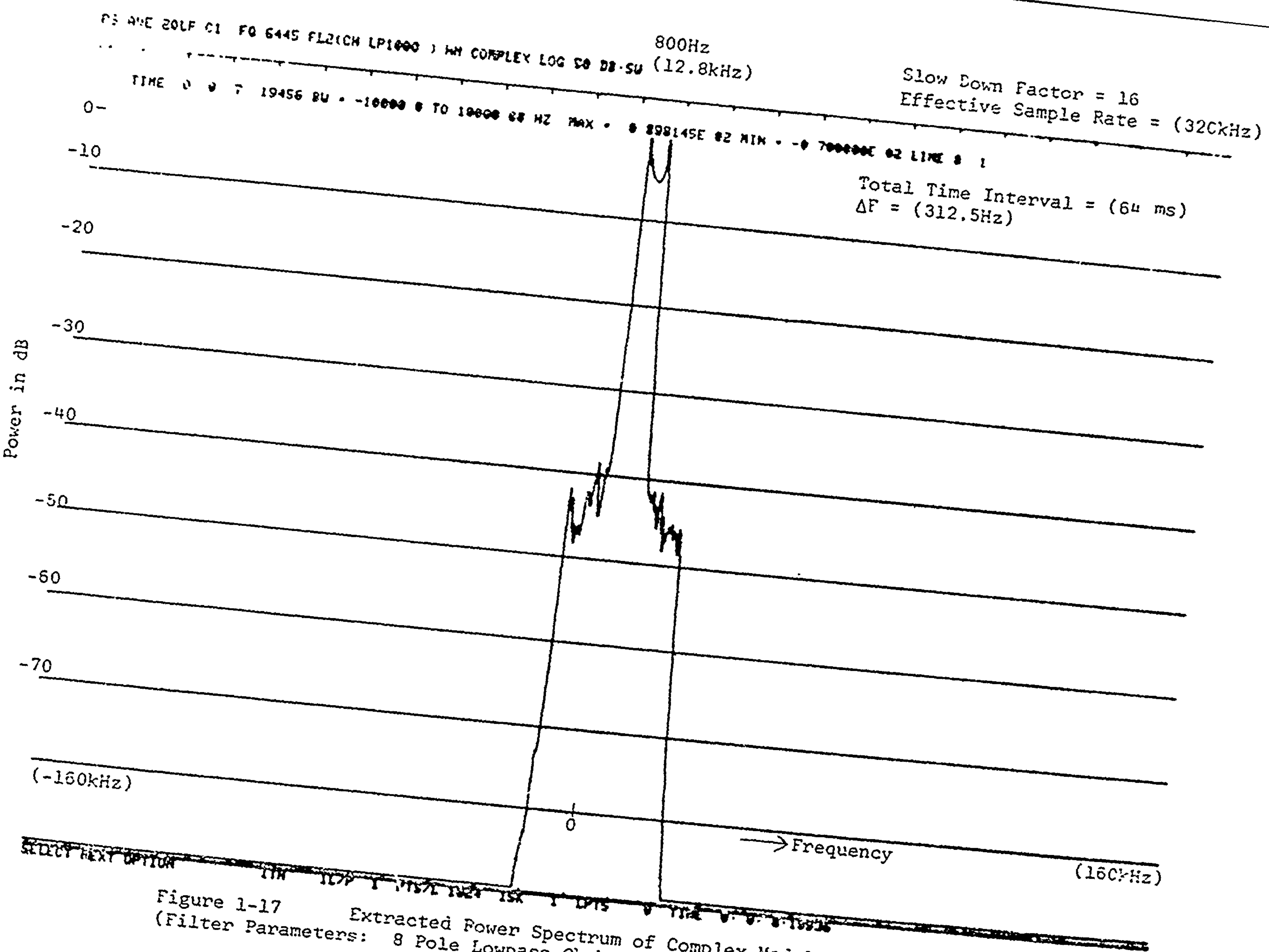


Figure 1-17 Extracted Power Spectrum of Complex Modulation Envelope
(Filter Parameters: 8 Pole Lowpass Chebyshev with Cutoff Equal to 1000Hz)

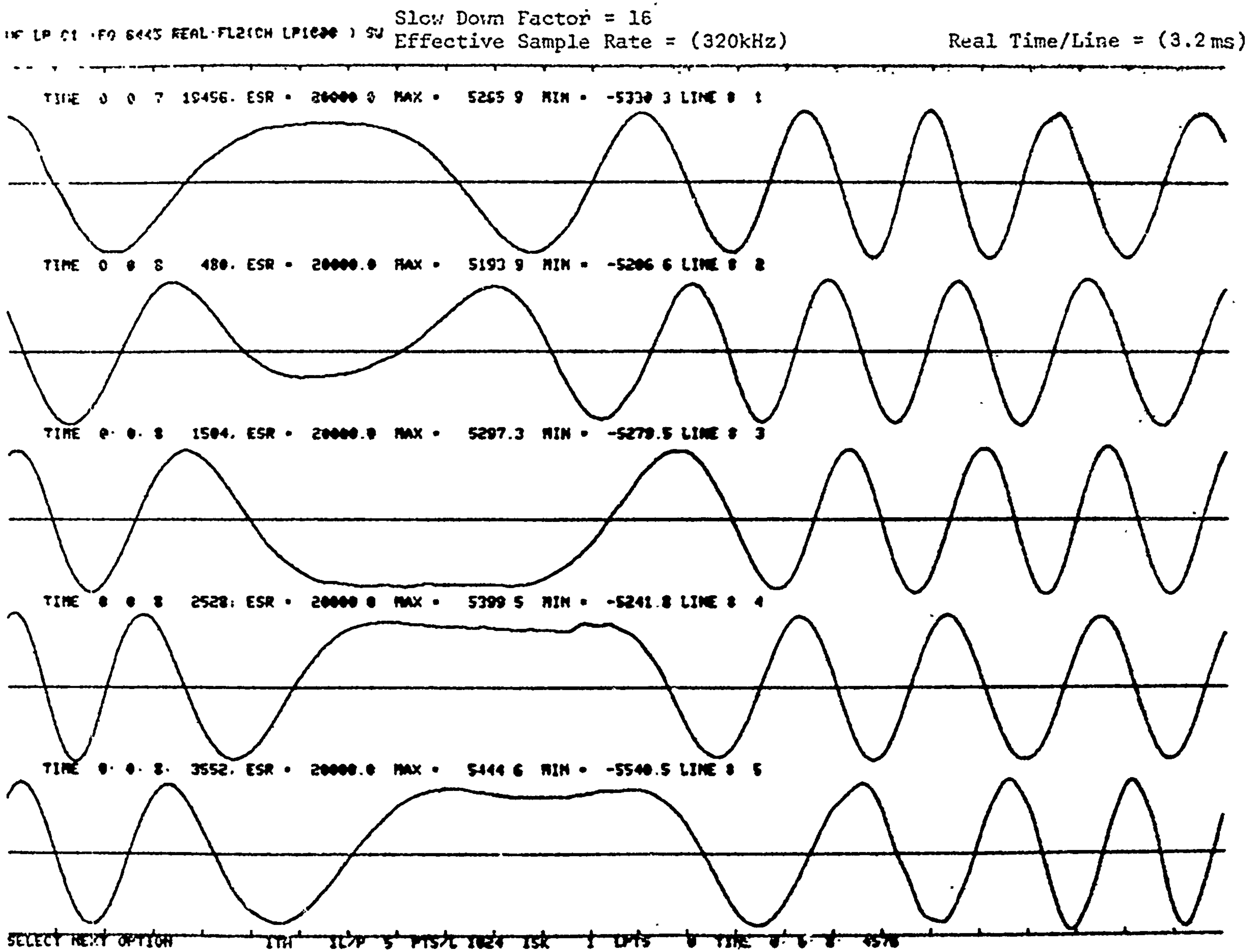


Figure 1-18 In-Phase Waveform Plot

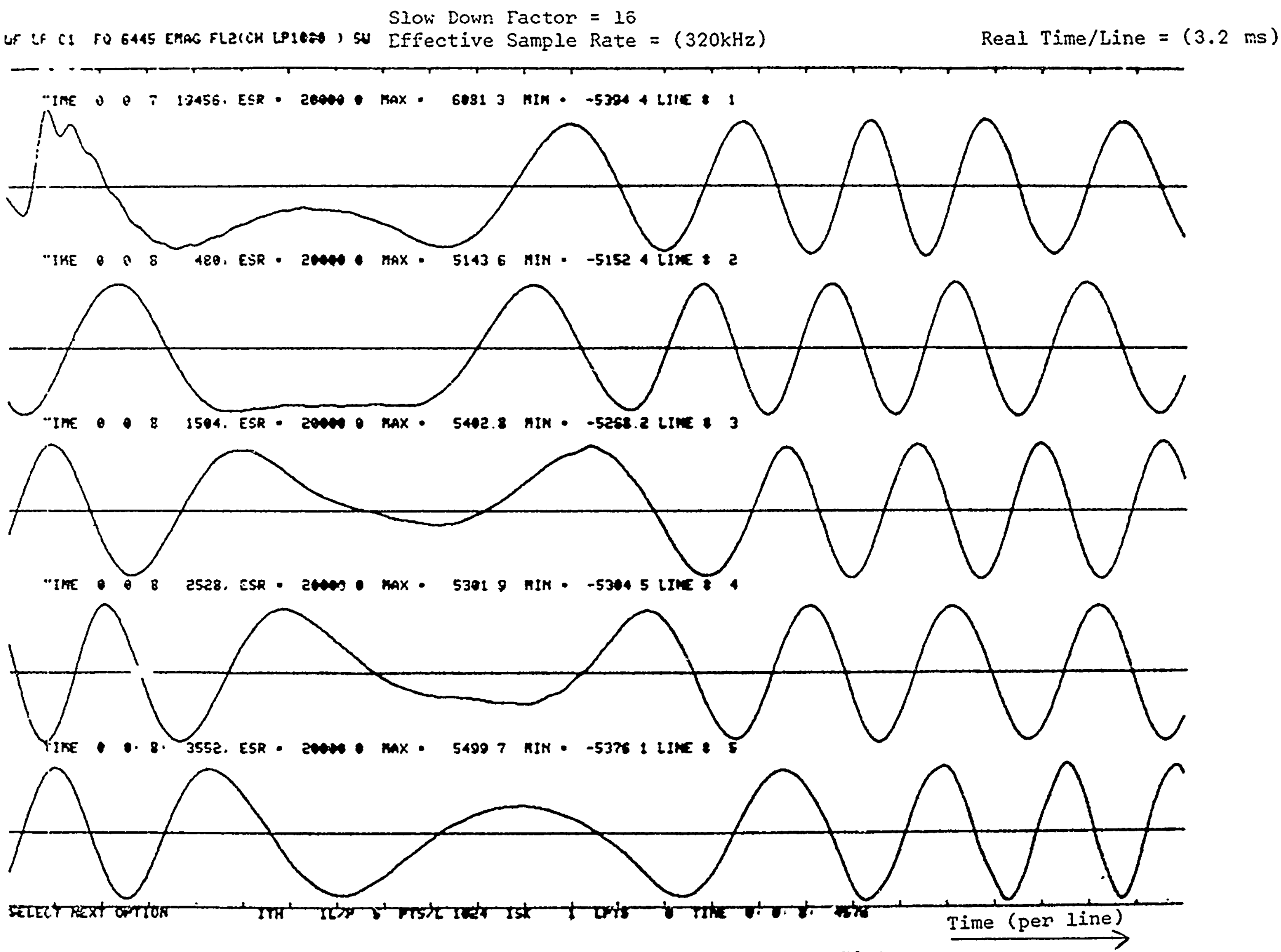


Figure 1-19 Quadrature Phase Waveform Plot

1.4.8. Spectrum of Complex Modulation Signal after Resampling

Since the complex modulation signal that results from the process of frequency translation and filtering is well over-sampled, its sampling rate can be lowered without introducing aliasing distortion. A lower sampling rate can achieve economy in subsequent digital processing such as demodulation and is required for accomplishing a higher resolution spectral analysis.

Figure 1-20 shows a PSD of the resampled complex basebanded waveform. The resampling was accomplished by retaining every tenth sample, hence, a resampling factor of 10. The computer annotation "RS10" indicates the selected resampling operation.

The lower sampling rate is made possible by signal band limiting as shown in Figure 1-17. The increase in resolution of the PSD plot obtained by the resampling is evident by comparing Figures 1-17 and 1-20. Figure 1-21 shows a portion of the "before resampled" complex signal spectrum with an expanded frequency scale. The two plots show graphically the difference between PSD analysis with higher frequency resolution versus a PSD plot with an expanded frequency scale.

The expanded frequency scale PSD plot shown in Figure 1-21 was obtained by the user inputting desired values for "Ith" and "Jth."

1.4.9. AM and FM Waveform Plots

Figures 1-22 and 1-23 show respectively the AM and FM waveforms that result from the digital demodulation of the resampled complex modulation waveform. Both plots have local line (LL) scaling.

The beginning of the first line of both the AM and FM plots shows the transient resulting from initiating the quadrature detection filter. The transients could have been eliminated from the data at the indicated page-starting time by having initiated the filter at an earlier time.

The computer display annotation "DEMOD:AMP" indicates the AM waveform, and "DEMOD:DPHASE/DT" indicates the FM waveform. MAX and MIN are referenced in terms of % modulation in the case of the AM waveform. FM waveforms are referenced in terms of Hz deviation.

The nearly sinusoidal squelch side-tone FM modulation is clearly revealed in Figure 1-23. A "zero" frequency line representing the shift or synchronous frequency is shown in conjunction with each of the five lines of data. It is observed that the side-tone modulation dc-level lies slightly below the zero frequency line. This displacement represents the slight error in selecting the "shift" frequency used in accomplishing the quadrature detection.

In addition to the side-tone modulation, bursts of high frequency noise are also observed in the FM waveform. Calibrations have shown that these

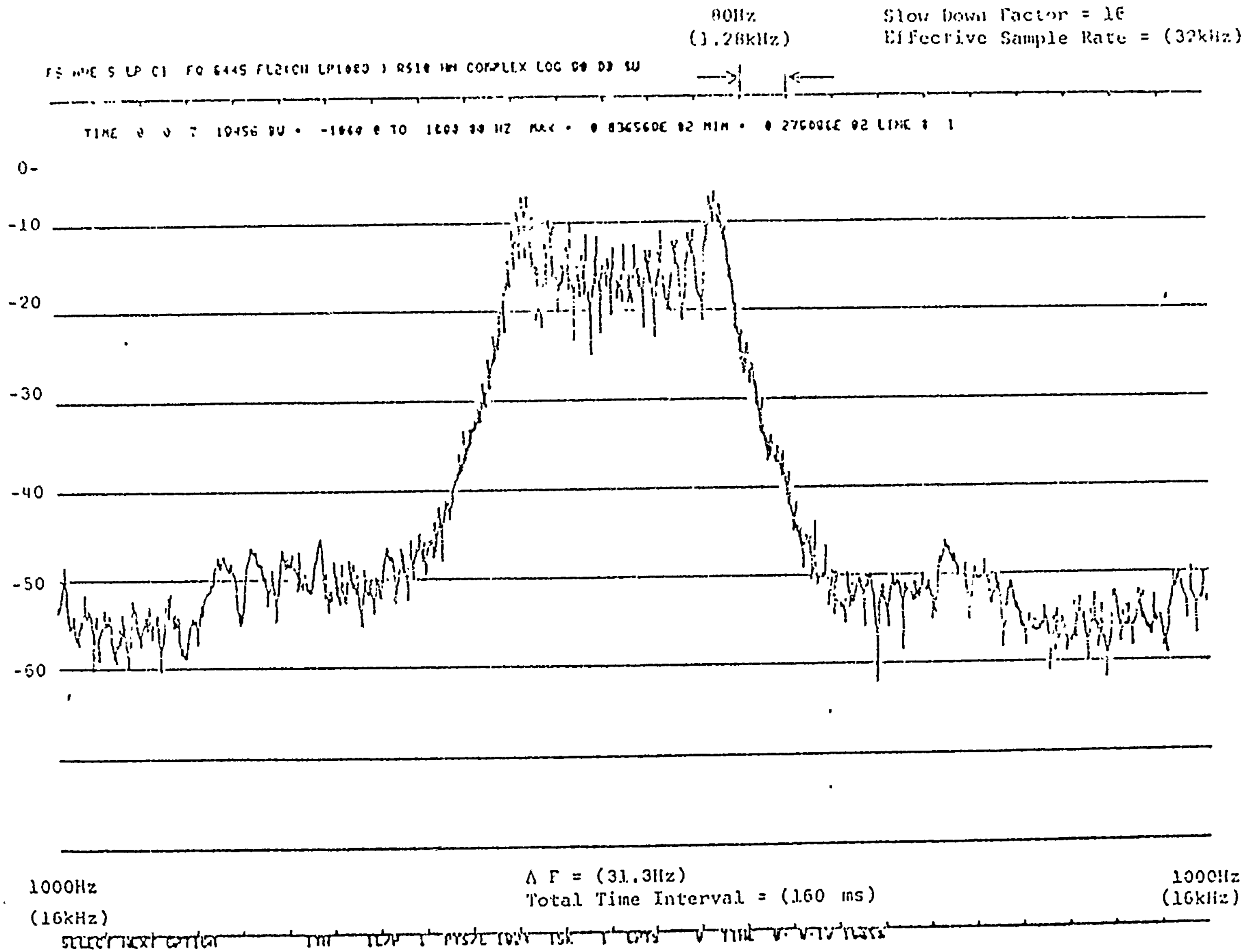


Figure 1-20 Power Spectrum of Extracted Complex Modulation Envelope After Resampling by a Factor of 10

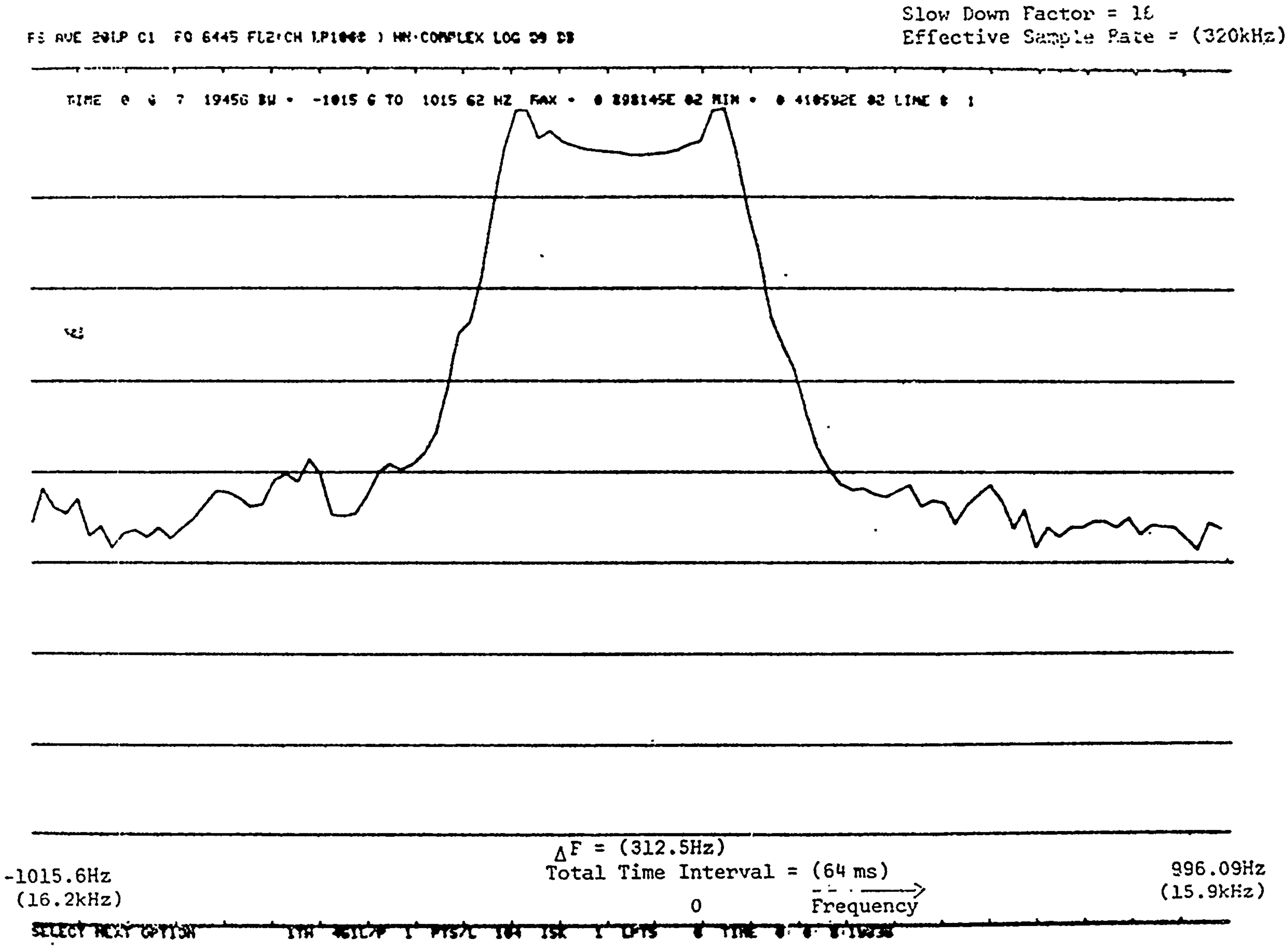


Figure 1-21 Power Spectrum of Extracted Complex Modulation Envelope With an Expanded Frequency Scale

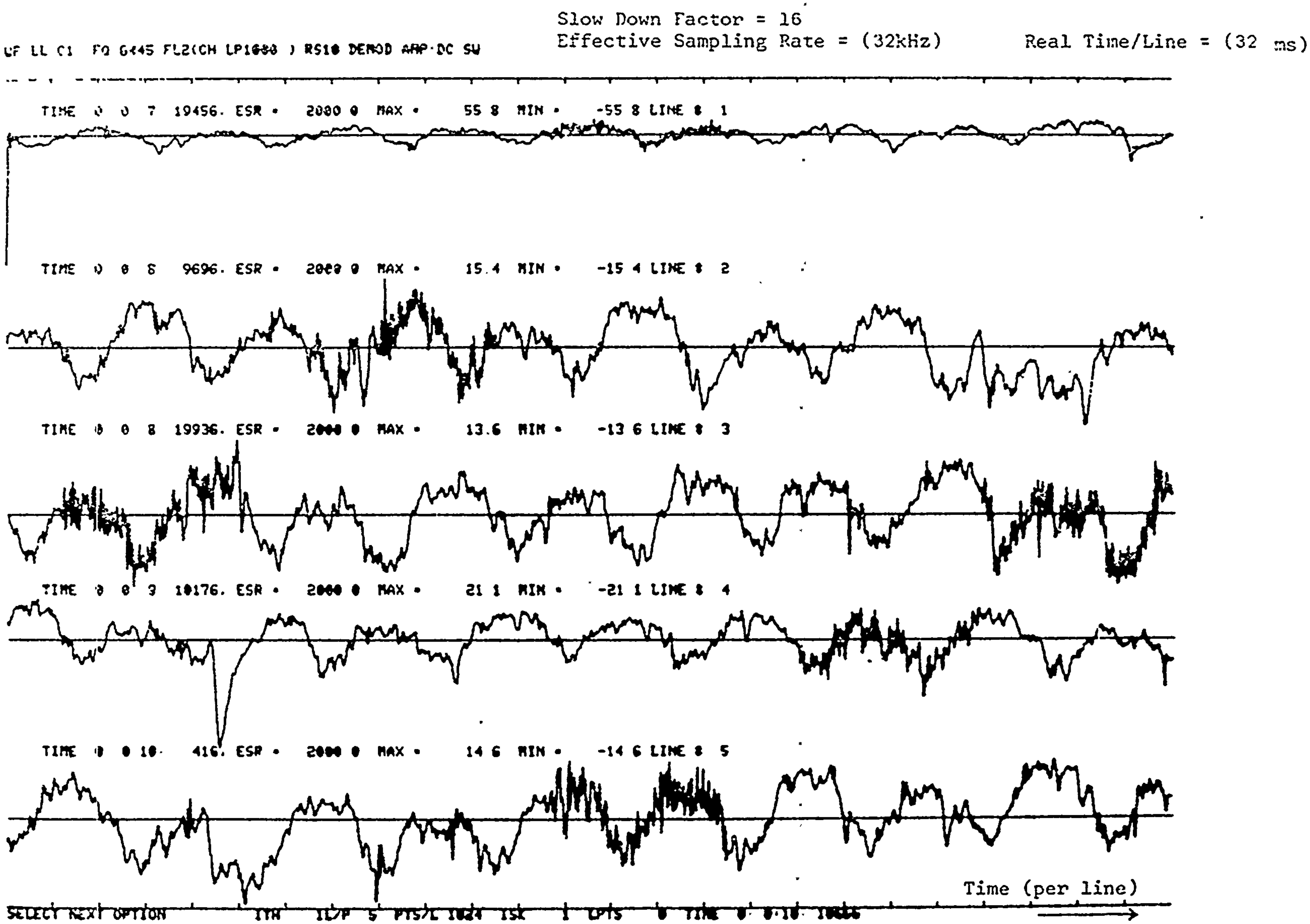


Figure 1-22 Extracted AM

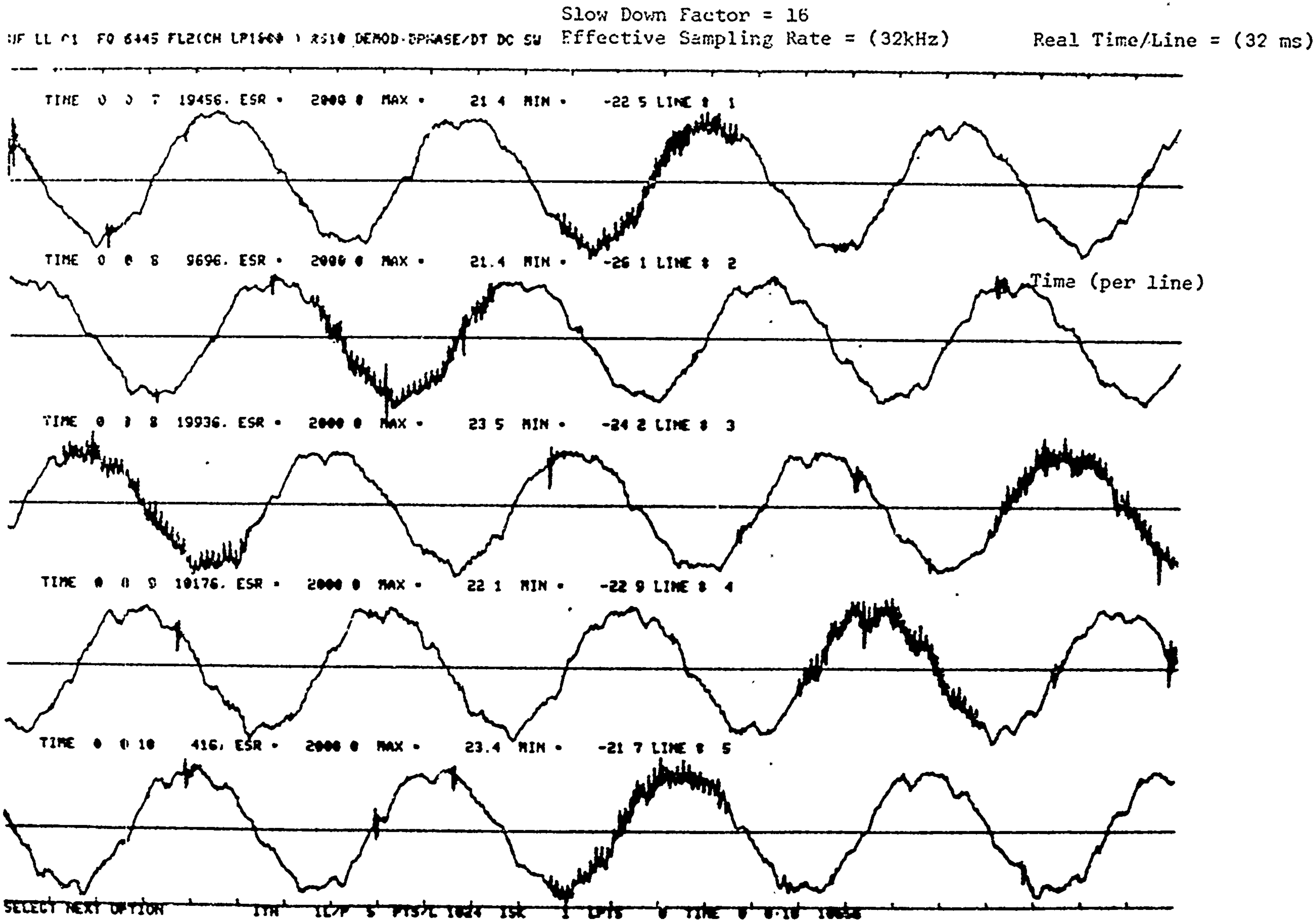


Figure 1-23 - FM Waveform

noise bursts were introduced by a malfunctioning analog-to-digital converter (ADC). Currently, digitizing is being done at PAR's ADC facility.

The squelch side-tone modulation parameters can be estimated from the FM waveform plot. The rate of FM is calculated first. There are 24 cycles of the modulation shown and these occur during an interval of 0.16 seconds (signal time). Hence, the rate of modulation,

$$\text{rate} = \frac{24 \text{ cycles}}{0.16 \text{ sec}} = 150 \text{ Hz.}$$

The peak-to-peak frequency deviation is calculated next. The waveform "MIN" and "MAX" values indicated on Figure 1-23 are in Hz deviation referenced to digitizing time. To compute the real-time deviation we multiply by 16 or

$$f_{pp} = 16(\text{MAX-MIN})$$

where f_{pp} is the peak-to-peak frequency deviation.

It is noted that the factor 16 in the above calculation accounts for the scaling required because of 16 times tape play back slow-down. Hence, the peak-to-peak frequency deviation is equal [2924 - (-3911)]Hz or 6835 Hz.

Both the FM rate and peak-to-peak frequency deviation measurement values are reasonable for the RT-524 squelch modulation.

The AM waveform plotted in Figure 1-22, also reveals a periodic waveform, although embedded with much more noise than in the FM waveform. Inspection of the periodic AM component shows that its modulation rate is twice that of the FM side-tone modulation. This AM is a cross-modulation from FM that occurs as an unintentional property of the RT-524 transceiver. It is noted that the bursts of high frequency noise observed in the FM waveform are also evident in the AM waveform. The annotation "MAX" and "MIN" show that the average % AM modulation over one line is approximately 15%.

Finally, it is noted that both the FM and AM waveform plots were obtained from resampled waveforms, where the resulting effective sample rate was 32K samples/sec. The waveforms with this sample rate appear sufficiently over-sampled for analysis.

1.4.10. FM and AM PSD Plots

Figures 1-24 and 1-25 show respectively plots of FM and AM power spectral density (PSD) on a dB scale. Max and Min are referenced in dB relative to 1% modulation for the AM spectrum. The FM spectrum is referenced in dB relative to 1Hz deviation.

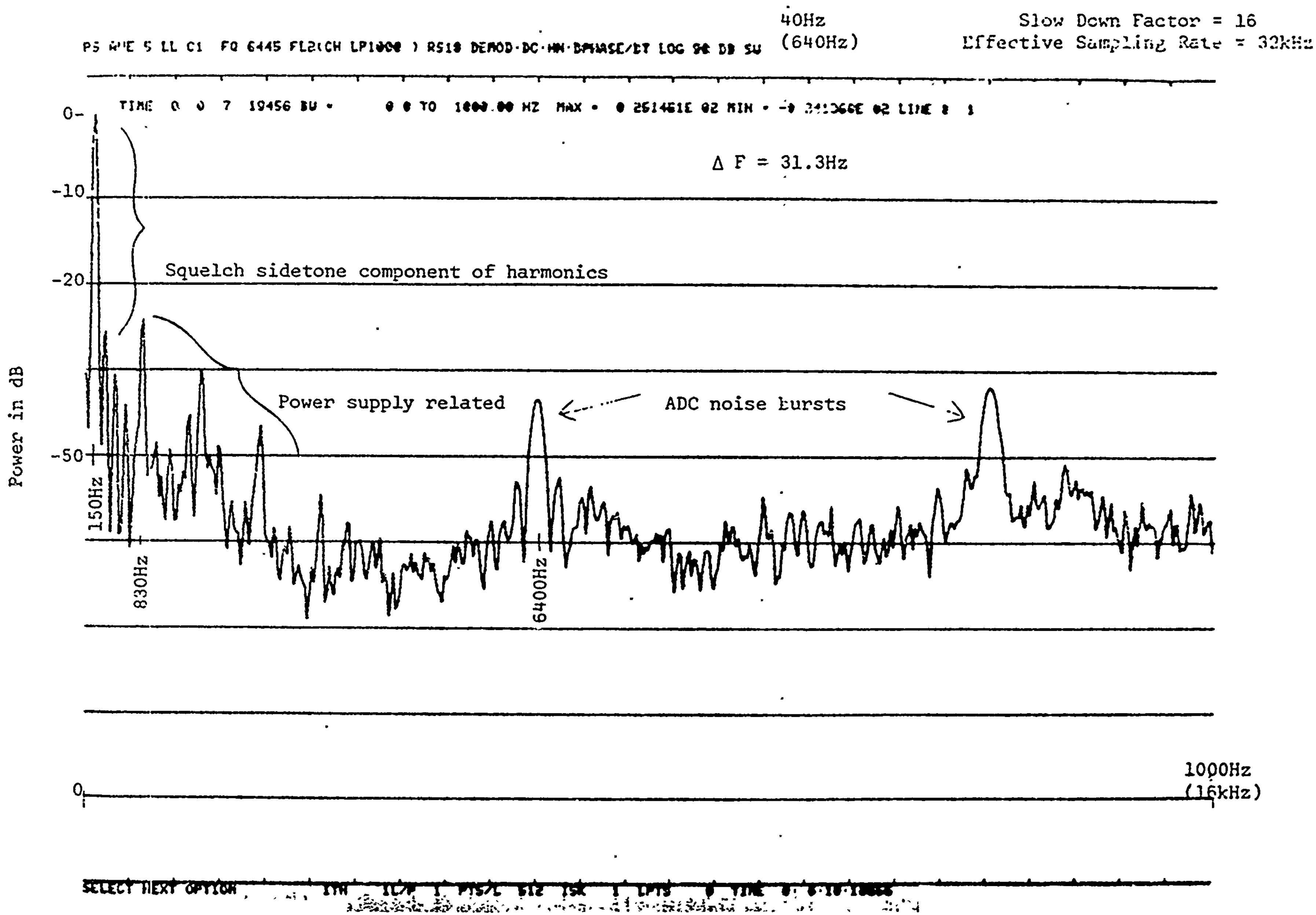


Figure 1-24 Power Spectrum of Extracted FM Waveform

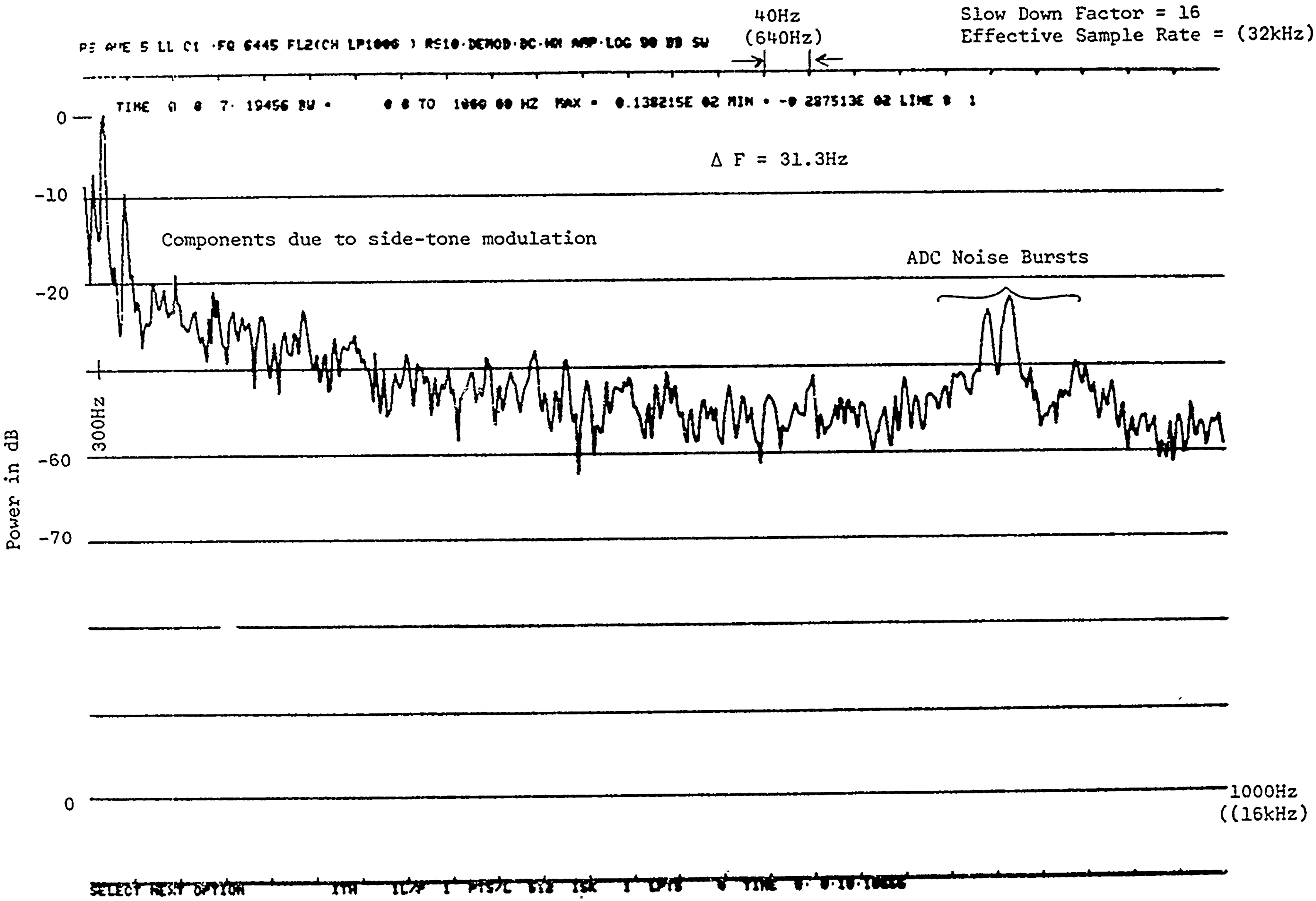


Figure 1-25 Power Spec um of Extracted AM Waveform

In the FM PSD the squelch side-tone modulation component at 150 Hz is clearly dominant, at least 25 dB above other components. Harmonics of the side-tone modulation are also evident, with the second harmonic approximately 27 dB below the fundamental.

A second coherent component of the FM PSD appears at a frequency of approximately 830 Hz. This component, which also appears with harmonics, is believed to be associated with the RT-524 power supply.

Two additional major spectral components are observed. These are a component centered at 6400 Hz and its harmonic centered at 12,800 Hz. These components, which are wide relative to the previously discussed components, are due to the ADC produced noise bursts observed in the FM waveform. The fact that the spectral lobes are concentrated indicates that the "noise" within the bursts contains an approximately periodic waveform component at a fundamental frequency of 6400 Hz. The width of the observed spectral component is at least in part due to burst pulse-width and noise.

Finally, the random part of the FM PSD appears as an approximately "white" level, down 50 dB or greater from the squelch modulation.

The AM PSD plot shows a relatively high low-frequency component. This is due to the large dc component of the AM waveform. The value at "zero" frequency has not been plotted so that the remainder of the PSD can be displayed with a display dynamic range that is reasonable for observation (IDRS gives the user the option of removing a waveform dc value, before calculating a PSD plot).

Coherent spectral components associated with the FM side-tone modulation are clearly evident, with a 300 Hz second harmonic approximately 8 dB below the 150 Hz fundamental.

Coherent spectral components associated with the power supply are not evident in the AM PSD plot. Only a "second harmonic" ADC noise burst spectral component is evident centered at 12,800 Hz.

1.4.11. Waveform Histograms

An additional method for extracting information from raw data is the use of a histogram. We speculate that possible features might, for example, be indicated by the symmetry of a histogram, or by some special trait of the tails of a histogram.

IDRS has one such option - a histogram that displays frequency of occurrence vs. waveform amplitude - while a second histogram option is currently available independent of IDRS. This additional option is discussed later.

The IDRS option can generate a histogram of any waveform or power spectrum. Figure 1-26 shows a histogram of a waveform which oscillates between two values.

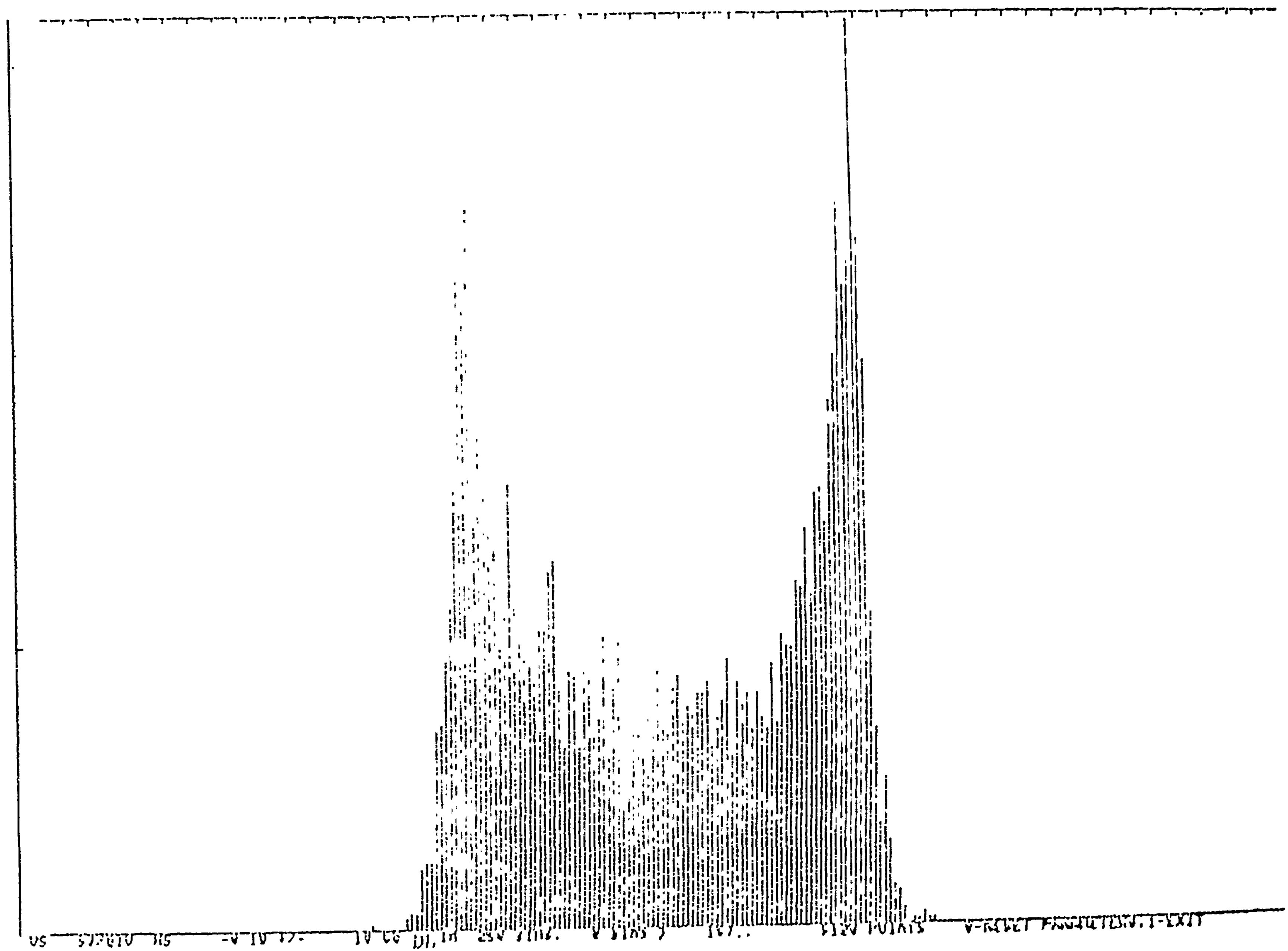


Figure 1-26 Histogram of FM Waveform

Referring to the bottom annotations, the "VS: 50/DIV" indicates that a bin containing 50 occurrences will have a height of one division on the vertical scale. The "HS: -0.19 +/- 10.00 DE" gives the horizontal scale: -0.19 as the mean value of the raw data used for this histogram, and is displayed in the center bin on the page; thus, the full range of the displayed data is -0.19 - 10.00 to -0.19 + 10.00, in units specified by the user. Here the letters "DE" mean "degrees." There are 250 bins across the page and 5120 total points displayed in the histogram. The number 157 indicates the maximum allowed bin size on the page, and is user-defined. Global scaling is also possible; i.e., scaling the histogram to its maximum value, as is the case here. The "0" is the number of bins whose value exceeded the maximum allowable. The user can continue to alter the various parameters until a histogram suitable to his liking is displayed. He can exit from this option by typing "1."

SECTION 2

IDRS, WPS, AND WAVES, A COMPARATIVE ANALYSIS

The Interactive Digital Receiver Simulator (IDRS), the Waveform Processing System (WPS), and the MULTICS Long Waveform Analysis System (WAVES) are each powerful waveform structure analysis and feature definition tools. Here, we will analyze each of the systems first from the standpoint of system design philosophy and user requirements, and secondly, from the macroscopic software point of view. Each of the systems presents favorable points. It is the purpose of this section to explore each of the systems in sufficient detail so as to indicate to an interested party whether or not the particular problem he has at hand is suitable to any of the software systems.

2.1. DESCRIPTION OF THE DESIGN PHILOSOPHY AND USER REQUIREMENTS

The philosophy by which a waveform structure analysis/feature definition system was designed will largely determine the nature of pattern recognition problems to which it is likely to apply. After the identification system capabilities, we shall examine the requirements of the user for each system.

The following three subsections present a discussion of the design philosophy and user requirements for the WPS system, WAVES system and the IDRS system. Finally, the fourth subsection presents a comparative analysis of the three systems.

2.1.1. Description of the Design Philosophy and User Requirements of WPS

The WPS system is a complete stand-alone dedicated waveform feature definition/extraction and classification system. The design of the WPS system is such as to allow the entry of raw waveform data trees and to permit complete waveform structure analysis. When promising features are determined, a waveform feature extraction algorithm can be implemented using the PARLAN language within WPS. The language provides simplified waveform and vector data structures and I/O, as well as a powerful library of PARLAN subroutines available for user program support. PARLAN program editing, compilation and execution is performed within WPS. After the waveform features have been extracted, they can immediately be examined in the OLPARS (On-Line Pattern Analysis and Recognition) portion of the WPS system. The general design philosophy of the WPS system was to provide a means for the user to perform waveform structure analysis, hypothesize and explore data transformations to identify potentially discriminating data characteristics, extract these discriminating features, and analyze these feature vectors using OLPARS, all within the confines of a single system.

The WPS system accepts data from magnetic tape or cards and stores it in a tree structure. A full complement of file and tree manipulation and editing utilities have been implemented under WPS. Statistical tabulation, single and multiple waveform editing and segmentation are

permitted. The wide variety of waveform transformations available includes Walsh/Hadamard transformations, algebraic calculus, eigenvectors, discriminate vectors, digital filtering, spectral analysis, and cepstrum.

The WPS system is quite receptive to the novice user. The logical frame hierarchy is easily understood and manipulated by the new user. Since WPS has no operating system, it is a complete system unto itself, with a consistent command format and error messages tailored to the WPS user. Also, there is no requirement for the user to master the WPS system and also master the host operating system. WPS does, however, demand that the pattern analysis and recognition language (PARLAN) be used for the development of feature extraction algorithms. The PARLAN language is similar in structure to Fortran with subroutine capability, but also provides a generalized I/O scheme tailored for waveform and vector data.

The WPS system boasts enhanced graphics display capabilities with a high speed Vector General (VG) random scan graphics terminal. Along with the terminal keyboard; a light pen, graphics tablet, sixteen button function box, and eight unit potentiometer box are all used as input devices to the system. The VG terminal can display several thousand vectors and can perform blinking and elimination of display components. Signals may be compared by enabling a shift option which permits the user to slide the waveform up and down the screen for comparison against another.

The documentation on the WPS system is quite complete. A potential user who can digest the operator's manual for the WPS will confront few problems running the system. Control traps and software error recovery software is standard throughout the system. WPS has no on-line help except for the available menus. All WPS command parameters are resolved in a query session after the particular option has been selected. Because of this, the user is constantly aware of exactly what the system is doing. The more advanced user, however, may find this scheme time-consuming.

2.1.2. Design Philosophy and User Requirements for WAVES

WAVES (the Multics Long Waveform Analysis System), is a waveform feature definition and extraction system which serves as a front-end to the Multics version of OLPARS (On-Line Pattern Analysis and Recognition System). Both OLPARS and WAVES are available through the ARPA computer network.

The WAVES system is designed to provide a capability to perform waveform feature investigation on multiple waveform data trees. A highlight of the system is the capability of multiple line waveform displays. The display can show consecutive waveform segments, segments of several nodes of the current data tree, a segment from each node of a tree, and waveforms or segments from different trees. The system provides extensive data tree manipulation utilities. Once a waveform transformation is defined and evaluated on a few test waveforms, the transformation can be easily applied to the entire data tree, resulting, typically, in the creation of a new transformed data tree. Further, data tree transformations are strung together automatically through the use of a data set stack. As an option is run, it operates on the top data set of the stack (the current data set) and if a transformed tree is produced, the stack is pushed and the new transformed data set is placed in the position of the current data set. Therefore, the next option run would interpret

the transformed data set as input. To facilitate the rerunning of options or experiments, WAVES provides a full set of data set stack manipulation commands. Although it is simple to string together a sequence of transformation options and operate on entire data trees with great ease, WAVES does not provide an extensive library of transformation algorithms.

The transformations available in WAVES are filtering, forward and reverse FFT, power and phase analysis, and a variety of waveform segmentation options. The design philosophy of WAVES is to provide an efficient means of multiple waveform data base management together with standard feature definition primitives and a means for the user to input his own specialized feature definition algorithm as his specific problem dictates.

The WAVES system allows the user to enter, compile, link and debug his own Fortran or PL/I algorithms, all within the confines of WAVES. Further, the extensive repertoire of WAVES data manipulation, display, and transformation routines are each available to the user's program in the form of a well-documented subroutine library. This capability provides the key mechanism for feature extraction as well as an important part of feature definition. It is unlikely that even the most comprehensive algorithm library will contain all the tools ever needed. In view of this, any feature definition/extraction system which is to remain useful must permit expansion (the inclusion of new algorithms). The philosophy of the WAVES system suggests that a waveform feature definition

analyst, in order to exercise all promising hypotheses, should be able to implement his own algorithms for testing. WAVES provides a work bench with a large supply of tools at the disposal of the analyst for tailored algorithm implementation and verification.

Although feature extraction with WAVES suggests a certain level of familiarity with the system, waveform structure analysis and feature investigation necessarily do not. In this regard, the interactive interface of the WAVES system is quite unique. There exist two distinct modes of WAVES command interaction. The "query" mode can be invoked spontaneously on any command by ammending the suffix "q" to the command. The inclusion of the "q" requests that the current command be interpreted in query mode. In query mode, each of the qualifiers of the command are individually prompted (usually with default values displayed) to the user. For each command qualifier the user must either select the default or enter a new value. Each of the parameters for each command are held in a common region initialized at system start-up. Parameters are updated directly in the common region and these new values become the system defaults. It is possible to enter any single command parameter directly without going through the command query session for each parameter of that command. This scheme permits the second mode of command interpretation. When a command is entered without the query suffix "q", the default values for the command parameters as stored in the common region are used. This provides the advanced user with the ability to modify those default parameters required and then invoke the desired

command in default mode. Such an invocation is very fast and rarely prompts for input at all.

For the novice user, a command summary is available on-line in the form of a menu. Also, WAVES provides fairly detailed explanations of each command in the form of an on-line "help" facility. The "help" command may be invoked with qualifiers requesting information on any of the major command categories. This feature was found particularly valuable to the novice user.

Any system available to the public, as WAVES is via the ARPA computer network, should meet certain requirements in its receptability to introductory users. The WAVES system, which utilizes English language commands, extensive (and documented) error information, on-line menu and help capabilities and a spontaneous "query" mode command processor, certainly surpasses these requirements. The novice user is able to perform common waveform transformations and general structure analysis. The application of WAVES to complex waveform feature definition/extraction and pattern recognition problems requires that the user be able to define and implement his own algorithm, often using WAVES as a "work bench" to handle the associated complex data tree manipulation required to drive the user's algorithm.

2.1.3. Design Philosophy and User Requirements of IDRS

IDRS was designed as a single waveform feature definition system with no direct feature extraction capabilities. The IDRS system is capable of simulating the general functions of a digital receiver. The system is structured in the form of a logical sequence of modules which can be selectively activated by the user to achieve a desired function. A display link allows the display of time or frequency domain waveforms at any point in the receiver simulator. An attractive design feature is the ability to easily exercise control over the waveform segments down to a single sample point. This, together with the extensive signal processing capability provided, makes the system very well suited to waveform structure analysis and feature definition problems.

IDRS is designed to provide the user with an easy and flexible means of simulating general receiver modules. A menu of the available options is available on-line which identifies a one or two character mnemonic to select a particular command option. For any required parameters for a selected command, the system will initiate a "query" session which prompts for each parameter. Since the system is single waveform oriented, the complex bookkeeping for data trees is not present, which greatly reduces the number of required commands.

IDRS is not currently available on the ARPA computer network. The system runs on DEC PDP-11's and VAX computers under the RSX 11-M and VAX/VMS operating systems, respectively. Waveform identification is performed via the standard operating system file name conventions. Any number of waveforms (subject to storage limitations) may be maintained on the system for processing with IDRS one at a time. Since waveform file manipulation (creation, copying, deletions) are handled external to IDRS, it is expected that the user have a basic knowledge of the operating system.

For the advanced user who is familiar with the host operating system, it is possible to run IDRS in a batch mode for multi-waveform (data tree processing). To set up this mode of operation, the user would first select a series of options which define the process he wishes to run on the data. Using the host computer's Indirect Command File Processor, a list of waveform files can be sequentially processed using the previously determined IDRS command option list. The output from each run could be a series of graphic terminal hardcopies automatically generated, a transformed version of each waveform or waveform segment written either to disk or mag tape, or both. For example, it is possible to demodulate a waveform and create waveform files of the AM and FM components.

In order for a user to effectively use IDRS, he should have a general understanding of signal processing. It is possible for the

novice user to perform simple general time or frequency domain data analysis, but the use of IDRS as a receiver simulator requires an understanding of the generalized receiver structure. The menu of command options generally provides sufficient on-line help, and the system displays the current receiver configuration, parameters, and display options on each display page.

2.1.4. Comparative Analysis of Design Philosophy

The concepts of interactive pattern recognition problem solving were largely the same for each the IDRS, WAVES, and WPS system designs. Each system, however, plays a slightly different role in the problem solution.

The general form of the solution is:

- o waveform structure analysis/feature definition
- o waveform feature extraction
- o feature vector analysis using an OLPARS capability.

IDRS is proposed to meet the requirements for a comprehensive single waveform feature definition tool. It provides no direct integral feature extraction capabilities. Because of its specialized capacity, it is by far the simplest of the systems. This enables portability to other systems, and maintainability. Any or all of the receiver modules can be used, which permits the application of the IDRS to a large number of signal processing problems.

WAVES is designed as a feature definition/extraction system. WAVES represents the more elegant of the two tree structured data base systems (WAVES and WPS). The WAVES concept of standard input/standard output through the use of the current data set stack represents an elegant

means of relieving the user from cumbersome data tree manipulation between transformations. The design of the WAVES system allows for the inclusion of user-written algorithms in either Fortran or PL/I. WAVES invokes the standard MULTICS editor and compilers to perform development. Recall that in the WAVES system, the executive is MULTICS. Each option requested invokes a WAVES segment and returns control to MULTICS. For user task development and execution, all the support of the MULTICS system is available to the user. Further, the user can reference any WAVES command from within his program by referring to the equivalent subroutine call of the desired command. This provides the user with support for data tree manipulation, file access, transformations, etc. The effect of the user program is to use the standard input/standard output routines available in WAVES and appear as another WAVES command option.

The WPS system also represents a waveform feature definition and feature extraction capability. The WPS system is a single user system tailored for WPS operation. This is attractive when a dedicated capability is required for security considerations. Because it is a dedicated system, more data storage is available than is usually possible under WAVES.

WPS, like WAVES, is a data tree structured system. WPS runs stand-alone (it is its own operating system), and therefore provides its own files processor, overlay capability, executive, device drivers,

etc. Much of the WPS system, therefore, consists of its operating system. Without the support of the high level operating system, WPS options are often not as powerful as their WAVES counterpart, simply requiring more user interaction to reach the desired goal.

WPS was the longest in the making of the three systems. The transformation and general options available on the WPS system far surpass both IDRS and WAVES. IDRS provides specialized signal processing capabilities while WAVES relies mostly on user-written algorithms.

The feature extraction capabilities of the WPS system are dependent on the PARLAN language and its support. Because of the absence of a commercial operating system in WPS, it would be very difficult indeed to incorporate a standard higher level language (Fortran or PL/I). This necessitates the existence of PARLAN in WPS. If a standard higher order language is available, transportability of user algorithms is possible and generalized I/O can be achieved through calls to the software system waveform or vector data manipulation routines. This is the strategy used in WAVES.

Feature extraction using IDRS as a feature definition capability is accomplished by using IDRS to effect the predetermined transformations and store the result. Fortran routines are then implemented to operate on the transformed data. The IDRS primitives are available for use in the user's program. IDRS is a single waveform processing system.

The RSX 11-M or VAX/VMS operating system provides a means (the Indexed Command File Processor) of processing data trees sequentially through IDRS and the user feature extraction task. This is a form of batch processing which can run completely unattended.

2.2. MACROSCOPIC DESCRIPTION OF THE SOFTWARE SYSTEMS

In Section 2.1 the software systems were discussed from a philosophical and user's point of view. In the following sections we will examine the system from the standpoint of a macroscopic software overview. This discussion presents an overview of the components and structures of the various systems.

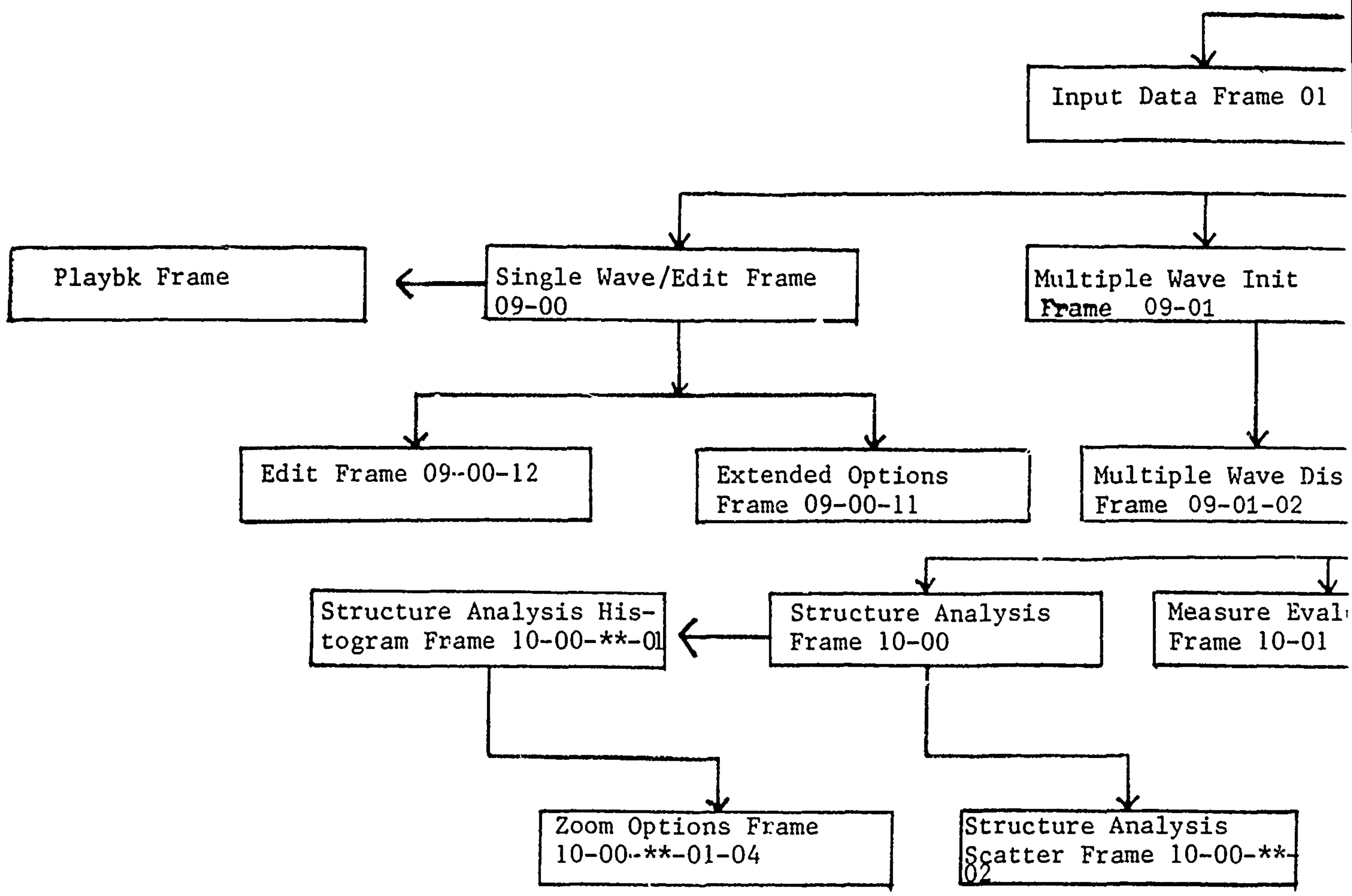
The following three sections describe WPS, WAVES and IDRS, respectively. Following the individual discussions will be a comparative analysis of all the systems presented in Section 2.2.4.

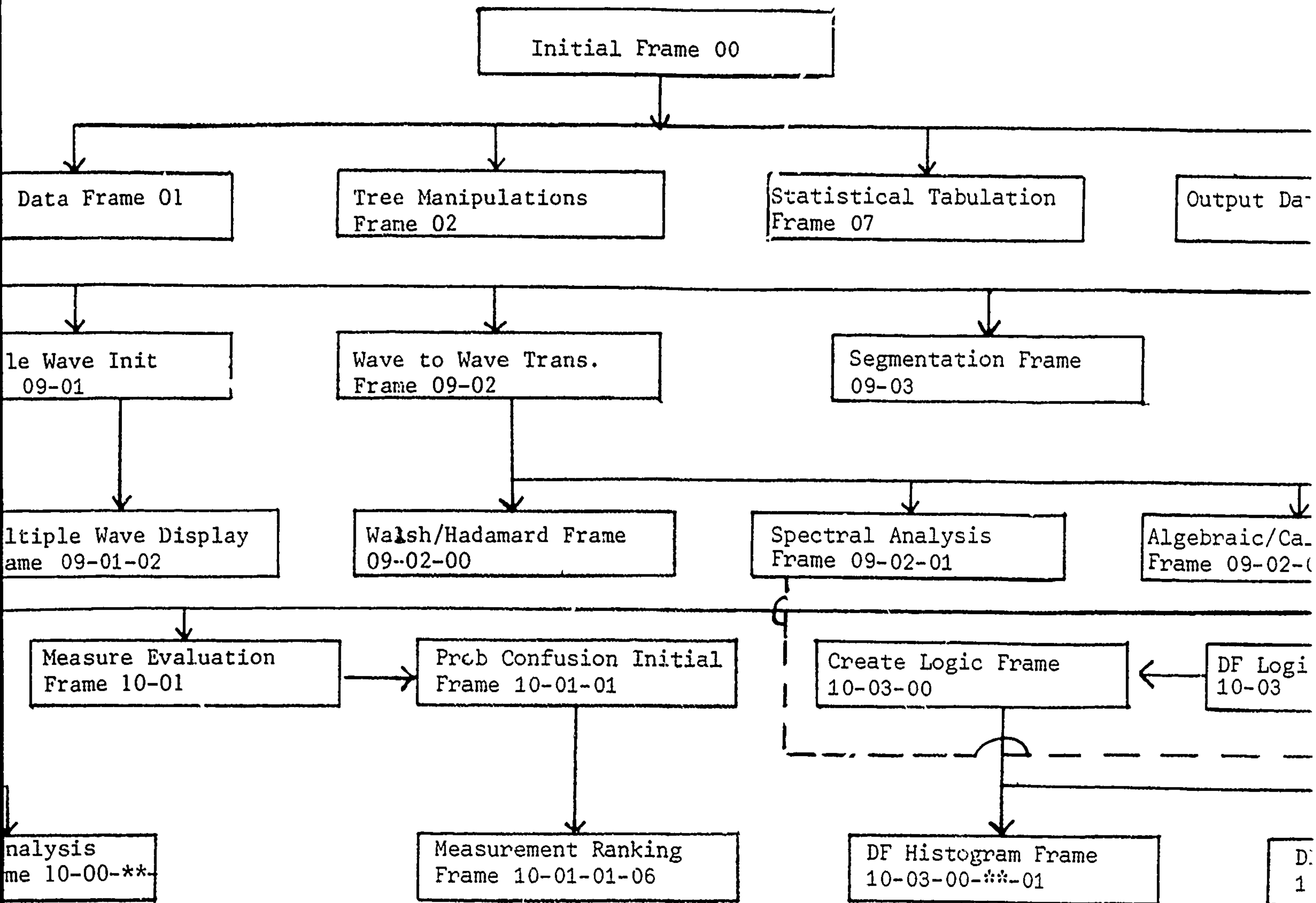
2.2.1. Macroscopic Description of the WPS Software

The WPS system is written entirely in assembly language. The programming of the WPS executive, filing and data management, and memory management for a specialized application provides minimum response time for the user. The applications algorithms of WPS take advantage of the Floating Point Processor in the PDP 11/45 in order to obtain the speed and accuracy required.

The commands are selected by the push button box switch which corresponds to the desired menu item. Query sessions are accomplished on the keyboard and the graphic tablet, light pen, and potentiometer box are used for graphic input.

Figure 2-1 shows the complete control tree for the WPS system. Examination of Figure 2-1, which shows more than 500 analyst selectable options, immediately demonstrates the complexity and sophistication of the system. Although the OLPARS segment is included in Figure 2-1, it will be discussed independently at a later point. The hardware configuration required for WPS support is shown in Figure 2-2. WPS is not a transportable package and because it maintains its own files processor, memory management and peripheral drivers, it is not likely that WPS can be run on alternate hardware configurations or supported by a high level operating system. Each of the WPS frames is a separate overlay. The overlays are manipulated by the core resident WPS executive. This


Input Data Frame 01
Playbk Frame
Single Wave/Edit Frame 09-00
Multiple Wave Init Frame 09-01
Edit Frame 09-00-12
Extended Options Frame 09-00-11
Multiple Wave Dis Frame 09-01-02
Structure Analysis Histogram Frame 10-00-**-01
Structure Analysis Frame 10-00
Measure Eval Frame 10-01
Zoom Options Frame 10-00-**-01-04
Structure Analysis Scatter Frame 10-00-**-02



Initial Frame 00
Data Frame 01
Tree Manipulations Frame 02
Statistical Tabulation Frame 07
Output Da-
le Wave Init 09-01
Wave to Wave Trans. Frame 09-02
Segmentation Frame 09-03
ltiple Wave Display ame 09-01-02
Walsh/Hadamard Frame 09-02-00
Spectral Analysis Frame 09-02-01
Algebraic/Ca- Frame 09-02-
Measure Evaluation Frame 10-01
Prob Confusion Initial Frame 10-01-01
Create Logic Frame 10-03-00
DF Logi 10-03
nalysis me 10-00-**-
Measurement Ranking Frame 10-01-01-06
DF Histogram Frame 10-03-00-**-01
D 1

2 of 3

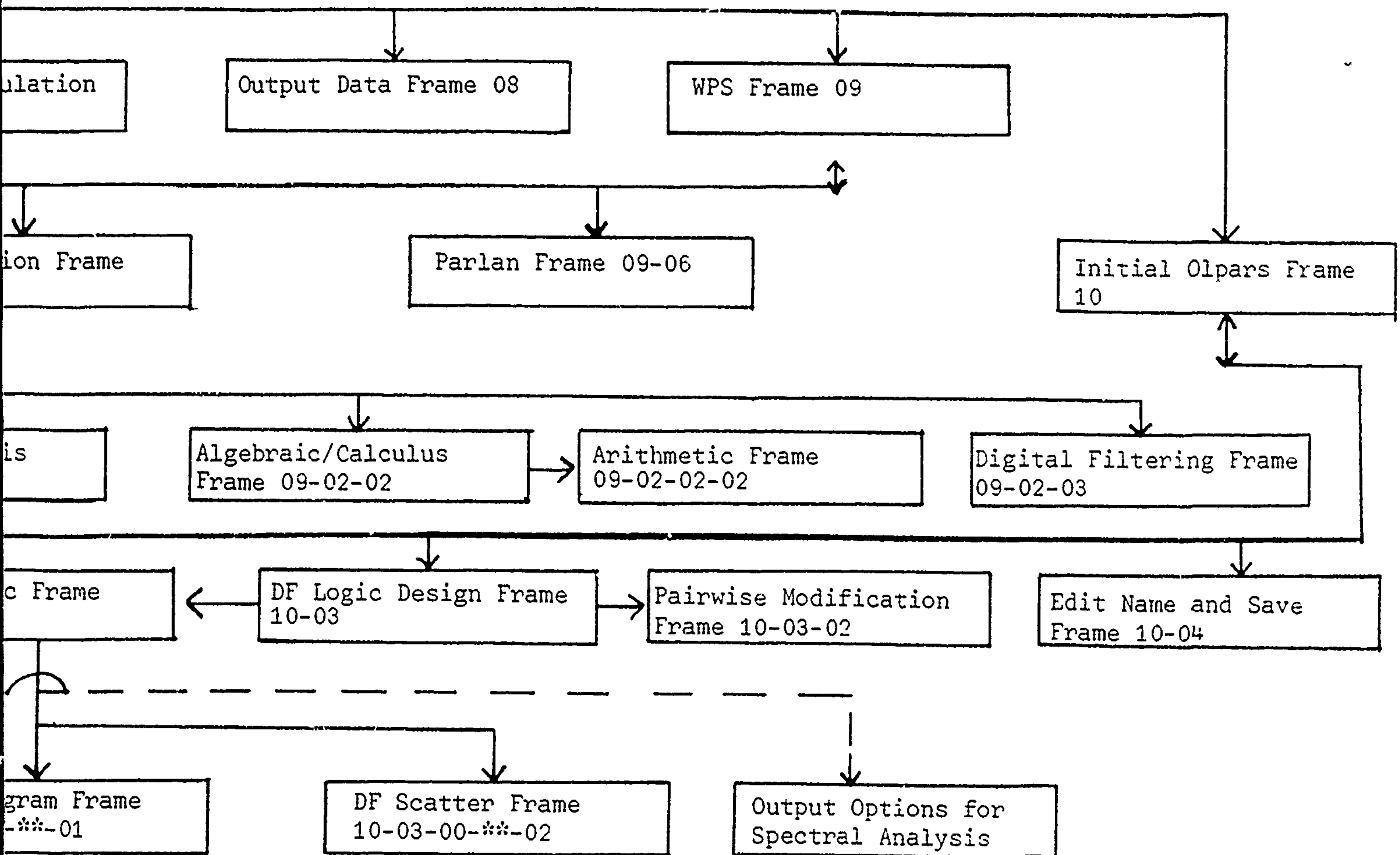


Figure 2-1 Options Available in the WPS System

Fig 2-1 (Cont'd on pages 2-19a thru 2-19m)

3 OF 3

INITIAL FRAME 00		INPUT DATA FRAME 01		TREE MANIPULATIONS FRAME 02	
00	INITIALIZE	00	WAVE TAPE INPUT	00	LIST DIRECTORY
01	INPUT DATA	01	VECTOR TAPE INPUT	01	LIST TREES
02	TREE MANIPS.	02	VECTOR CARD INPUT	02	DRAW TREE
03	LIST DIRECTORY	03	RESTORE TREE	03	DELETE TREE
04	LIST TREES	04	NAME AND SAVE	04	DELETE NODE
05		05	MODIFY TREE TEXT	05	DELETE SUBSTRUC
06	DRAW TREE	06		06	DELETE VECTOR(S)
07	STAT TAB	07		07	COMBINE NODES
08	OUTPUT DATA	08		08	CREATE TREE
09	WPS	09		09	APPEND NODES
10	OLPARS	10		10	APPEND FEATURES
11		11		11	SORT
12		12		12	CREATE DATA SET
13		13		13	SELECT DATA SET
14		14		14	RETURN TO INIT FRAME
15	HARDCOPY	15		15	HARDCOPY

OUTPUT DATA FRAME 08

00 PRINT DIRECTORY

01 PRINT TREES

02 PRINT TREE TABLE

03 PRINT HEADER INFO

04 PRINT SINGLE WAVE (VECTOR)

05 PRINT TREE/NODE

06 PRINT IDS

07 OUTPUT STANDARD WPS WAVEFORM TAPE

08 OUTPUT STANDARD VECTOR TAPE

09 SAVE ENTIRE TREE(S) ON TAPE

10 OUTPUT OLPARS LOGIC

11 LIST TREES

12 SELECT DATA SET

13 DRAW TREE

14 RETURN TO INIT FRAME

15

STATISTICAL TABULATION FRAME 07

00 VECTOR DATA RANGE

01 VECTOR DATA MEAN

02 VECTOR DATA DIF BETWEEN MEANS

03 VECTOR DATA VARIANCE

04 VECTOR DATA COVARIANCE MATRIX

05 VECTOR DATA CORR MATRIX

06 WAVEFORM DATA MEAN

07 WAVEFORM DATA VARIANCE

08 WAVEFORM DATA MIN/MAX

09 LIST TREES

10 SELECT DATA SET

11 DRAW TREE

12 GAUSSIAN NOISE GENERATION

13

14 RETURN TO INITIAL FRAME

15 HARDCOPY

WPS FRAME 09

00 SINGLE WAVE/EDIT

01 MULTIPLE WAVE

02 WAVEFORM TO WAVEFORM TRANSFORMATION

03 SEGMENTATION

04

05

06 PARLAN

07 PSEUDO 3D DISPLAY

08 STREAMING OPERATIONS

09 LIST TREES

10 SELECT DATA SET

11 DRAW TREE

12

13

14 RETURN TO INIT FRAME

15 HARDCOPY

PLAYBK FRAME		SINGLE WAVE/EDIT FRAME 09-00		MULTIPLE WAVE INIT FRAME 09-01	
00	PLAY THIS PART	00	DISPLAY NEXT	00	PAGE INIT
01	PLAY THIS WAVE	01	DISPLAY SAME	01	SCROLL INIT
02	PLAY A NEW WAVE	02	DISPLAY NAMED	02	MULTIPLE WAVE DISPLAY
03		03	DISPLAY PREVIOUS	03	TIME ALIGNMENT
04		04	HORIZONTAL SCALE	04	
05		05	VERTICAL SCALE	05	
06		06	CONTINUE WAVE	06	
07		07	SCROLL	07	
08		08	SELECT SEQUENCE	08	
09		09	ZOOM	09	LIST TREES
10		10	SELECT START TIME	10	SELECT DATA SET
11		11	EXTENDED OPTIONS	11	DRAW TREE
12		12	EDIT	12	
13		13	SELECT DATA SET	13	RETURN TO WPS FRAME
14		14	RETURN TO WPS FRAME	14	RETURN TO INIT FRAME
15		15	HARDCOPY	15	HARDCOPY

WAVE TO WAVE TRANS. FRAME 09-02

00 WALSH/HADAMARD
01 SPECTRAL ANALYSIS FRAME
02 ALGEBRAIC/CALCULUS
03 DIGITAL FILTERING
04 LATERAL GRIDS
05 AVERAGE VECTOR PER CLASS
06 CLUSTER CENTERS
07 EIGENVECTORS
08 DISCRIM-VECTORS
09
10 LIST TREES
11 SELECT DATA SET
12 DRAW TREES
13 RETURN TO WPS FRAME
14 RETURN TO INIT FRAME
15

SEGMENTATION FRAME 09-03

00 CREATE MARKERS
01 SEGMENT DATA
02 THRESHOLD OPTIMIZATION
03 BEGIN MARK
04 END MARK
05 CROSS CORRELATION AND CONVOLUTION
06 RISE OR FALL IN TIME WINDOW
07 AVERAGE POWER IN TIME WINDOW
08 AMPLITUDE LEVELS
09 ZERO CROSSINGS
10 LIST TREES
11 SELECT DATA SET
12 DRAW TREE
13
14 RETURN TO WPS FRAME
15 RETURN TO INIT FRAME

PARLAN FRAME 09-06

00 CARD INPUT
01 VG INPUT
02 EDIT PROGRAM
03 COMPILE PROGRAM
04 EXECUTE PROGRAM
05 DELETE PROGRAM
06 PRINT PROGRAM
07
08
09
10
11
12 RET TO WPS FRM
13 RET TO INIT FRM
14
15

INITIAL OLPARS FRAME 10		EDIT FRAME 09-00-12		EXTENDED OPTIONS FRAME 09-00-11	
00	STRUCT ANAL	00	CHANGE HEADER INFO	00	DISPLAY NEXT
01	MEASURE EVAL	01	MODIFY TEXT	01	DISPLAY SAME
02	TRANSFORMS	02	TRUNCATE END WAVE	02	CONTINUE WAVE
03	DF LOGIC DESIGN	03	TRUNCATE BEGINNING WAVE	03	LIST COORDINATES
04	EDIT NAME & SAVE	04	INSERT APRIORI SEGMENT MARKERS	04	DISPLAY DESCRIP LANG
05		05	SPECIFY TEMPORARY SYMBOL	05	DISPLAY CALCULATED MARKERS
06		06	SEGMENT WAVEFORM	06	DISPLAY APRIORI MARKERS
07		07	DELETE WAVEFORM	07	
08		08		08	
09		09	MODIFY TIME REFERENCE	09	
10		10	CONTINUE WAVE	10	
11	SELECT DATA SET	11	DISPLAY SAME	11	
12	LIST TREES	12	DISPLAY NEXT	12	
13	DRAW TREE	13	RETURN TO SINGLE WAVE/EDIT FRAME	13	RETURN TO SINGLE WAVE
14	RET INIT FRM	14	RETURN TO WPS FRAME	14	RETURN TO WPS
15	HARDCOPY	15	HARDCOPY	15	HARDCOPY

MULTIPLE WAVE DISPLAY FRAME 09-01-02

00 SELECT SEQUENCE

01 DISPLAY NEXT (ALL)

02 DISPLAY NEXT (ONE)

03 CONTINUE WAVES

04 CONTINUE WAVE

05 AUTO SCROLL

06 DISPLAY SAME

07 HORIZONTAL SCALE

08 VERTICAL SCALE

09 HORIZONTAL TRANSLATION

10 VERTICAL TRANSLATION

11 SELECT START TIME

12 SELECT DATA SETS

13 RETURN TO MULTIPLE WAVE INIT FRAME

14 RETURN TO WPS FRAME

15 HARDCOPY

WALSH/HADAMARD FRAME 09-02-00

00 NO. OF POINTS

01 OUTPUT OPTIONS

02 FORWARD FWT

03 INVERSE FWT

04 ZERO FILL

05 MODIFY TEXT

06

07

08

09

10 LIST TREES

11 SELECT DATA SET

12 DRAW TREE

13 RETURN TO WAVE TO WAVE TRANSFORMATION FRAME

14 RETURN TO WPS FRAME

15 HARDCOPY

SPECTRAL ANALYSIS FRAME 09-02-01

00 OUTPUT

01 INPUT DATA SET

02 EVERY PTH WAVE

03 EVERY NTH PT

04 WINDOW PARAMETER

05 WEIGHTING

06 WINDOW FILL

07 CHANGE TREE NEXT

08 EXECUTE

09 LIST TREES

10

11 WAVE TO WAVE

12 SINGLE

13 MULTIPLE WAVE

14 BACK TO WPS FRAME

15 HARDCOPY

ALGEBRAIC/CALCULUS FRAME 09-02-02

00 NORMALIZE
01 SMOOTHING
02 ARITHMETIC
03 DEMODULATION
04 RECTIFICATION
05 INDEFINITE INTEGRAL
06 DIFFERENCE
07 EXPONENTIAL
08 LOG (NATURAL)
09 MODIFY TEXT
10 LIST TREES
11 SELECT DATA SET
12 DRAW TREE
13 RETURN TO WAVE TO WAVE TRANSFORMATION FRAME
14 RETURN TO WPS FRAME
15

ARITHMETIC FRAME 09-02-02-02

00 ADDITION
01 SUBTRACTION
02 MULTIPLICATION
03 DIVISION
04 MODIFY TEXT
05 LIST TREES
06 SELECT DATA SET
07 DRAW TREE
08
09
10
11
12
13 RETURN TO ALGEBRAIC/ CALCULUS FRAME
14 RETURN TO WAVE TO WAVE TRANSFORMATION FRAME
15

DIGITAL FILTERING FRAME 09-02-03

00 INPUT OPTIONS
01 BUTTERWORTH
02 CHEBYCHEV..
03 IDEAL FFT FILTERING
04
05
06
07
08
09
10 LIST TREES
11 SELECT DATA SET
12 DRAW TREE
13 RETURN TO WAVE TO WAVE TRANSFORMATION FRAME
14 RETURN TO WPS FRAME
15 HARDCOPY

STRUCTURE ANALYSIS HISTOGRAM FRAME 10-00-**-01

00 SELECT CLASSES
01 CHANGE BIN PARAMS
02 CHANGE PAGE SIZE
03 DISP BIN COUNT
04 ZOOM OPTIONS
05 COMPARE CLASSES
06 LOCAL VERT SCL
07 GLOBAL VERT SCL
08 NEXT CLASS
09 NEXT PAGE
10 NAME & SAVE
11 SET THRESHOLDS
12 PARTITION
13 SELECT BASIS
14 RET STRUC ANAL
15 HARDCOPY

STRUCTURE ANALYSIS FRAME 10-00

00 EIGENVECTORS
01 COORD VECTORS
02 GEN DISCRIM VECS
03 ARBITRARY VECS
04 FISHER VECTORS
05 NON-LIN MAPPING
06
07
08
09
10 SELECT DATA SET
11 LIST TREES
12 DRAW TREE
13 RET OLPARS FRM
14 RET INIT FRM
15 HARDCOPY

MEASURE EVALUATION FRAME 10-01	PROB CONFUSION INITIAL FRAME 10-01-01	CREATE LOGIC FRAME 10-03-00
00 DISCRIM MEASURE	00 CHANGE RANGE	00 EIGENVECTORS
01 PROB OF CONF	01 CHANGE BIN SIZE	01 COORD VECTORS
02 HISTOGRAM	02 CHANGE NUM BINS	02 GEN DISCRIM VECS
03	03 DISPLAY RANGES	03 ARBITRARY VECS
04	04 NEXT PAGE	04 FISHER PAIRWISE
05	05 PRINT DISPLAY	05 NEAREST MEAN
06	06 CONTINUE	06 BOOLEAN
07	07	07
08	08	08
09	09	09
10 SELECT DATA SET	10	10
11 LIST TREES	11	11
12 DRAW TREE	12	12 RET DF FRM
13 RET OLPARS FRM	13 RET MEA EVAL	13 RET OLPARS FRM
14 RET INIT FRM	14 RET OLPARS FRM	14 RET INIT FRM
15 HARDCOPY	15 HARDCOPY	15 HARDCOPY

DF LOGIC DESIGN FRAME 10-03	PAIRWISE MODIFICATION FRAME 10-03-02	EDIT NAME AND SAVE FRAME 10-04
00 CREATE LOGIC	00 ARBITRARY VECS	00 DISPLAY ENTRIES
01 EVALUATE LOGIC	01 FISHER VECS	01 INITIALIZE N & S
02 PAIRWISE MOD	02 BOOLEAN	02 DELETE VECTOR
03 CREATE BOOL REJ	03	03 PRINT VECTOR
04 CHG APRIORI PROB	04	04 KEYBD INPUT
05 DRAW LOG TREE	05	05 CARD INPUT
06 DELETE LOG TREE	06	06
07 DELETE LOG NODE	07	07
08 LIST LOG TREES	08	08
09 SELECT LOG TREE	09	09
10 PRINT LOG TREE	10	10
11 SELECT DATA SET	11	11
12 LIST TREES	12 RET DF FRM	12
13 DRAW TREE	13 RET OLPARS FRM	13 RET OLPARS FRM
14 RET OLPARS FRM	14 RET INIT FRM	14 RET INIT FRM
15 HARDCOPY	15 HARDCOPY	15 HARDCOPY

ZOOM OPTIONS FRAME 10-00-**-01-04

00	FULL RANGE
01	DIAL RANGE
02	TYPE RANGE
03	TYPE BIN RANGE
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	HARDCOPY

STRUCTURE ANALYSIS SCATTER FRAME 10-00-**-02

00	NEXT PAGE
01	CHANGE SCALE
02	PRINT ID'S
03	BLINK
04	ELIMINATE
05	NAME & SAVE
06	CLUSTER
07	ZOOM
08	DRAW BOUNDARY
09	PARTITION
10	SELECT BASIS
11	
12	
13	RET OLPARS FRM
14	RET INIT FRM
15	HARDCOPY

MEASUREMENT RANKING FRAME 10-01-01-06

00	RANK OVERALL
01	SEL/CLASS RANK
02	SEL/PAIR RANK
03	SEL/MEA RANK CLASS
04	SEL/MEA RANK PR
05	UNION BEST CLASS
06	UNION BEST PR
07	SELECT CUTOFF
08	SELECT ANX MEA
09	CREATE TREE
10	PRINT DISPLAY
11	
12	RET PROB CONF
13	RET MEA EVAL
14	RET OLPARS FRM
15	HARDCOPY

DF HISTOGRAM FRAME 10-03-00-**-01

00	SELECT CLASSES
01	CHANGE BIN PARAMS
02	CHANGE PAGE SIZE
03	DISP BIN COUNT
04	ZOOM OPTIONS
05	COMPARE CLASSES
06	LOCAL VERT SCL
07	GLOBAL VERT SCL
08	NEXT CLASS
09	NEXT PAGE
10	SET THRESHOLDS
11	EVALUATE
12	
13	SELECT BASIS
14	RET CRELOG FRM
15	HARDCOPY

DF SCATTER FRAME 10-03-00-**-02

00 NEXT PAGE

01 CHANGE SCALE

02 PRINT ID'S

03 BLINK

04 ELIMINATE

05 CLUSTER

06 ZOOM

07 DRAW BOUNDARY

08 EVALUATE

09

10

11 SELECT BASIS

12 RET CRELOG FRM

13 RET OLPARS FRM

14 RET INIT FRM

15 HARDCOPY

OUTPUT OPTIONS FOR SPECTRAL ANALYSIS

00 FFT

01 FFT MAGNITUDE

02 FFT PHASE

03 LOG FFT

04 INVERSE FFT

05 POWER

06 LOG POWER

07 CEPSTRUM

08 COMPLEX CEPSTRUM

09 COMPLEX CEPSTRUM MAG

10 COMPLEX CEPSTRUM PHASE

11 INVERSE CEPSTRUM

12 INVERSE FILTER

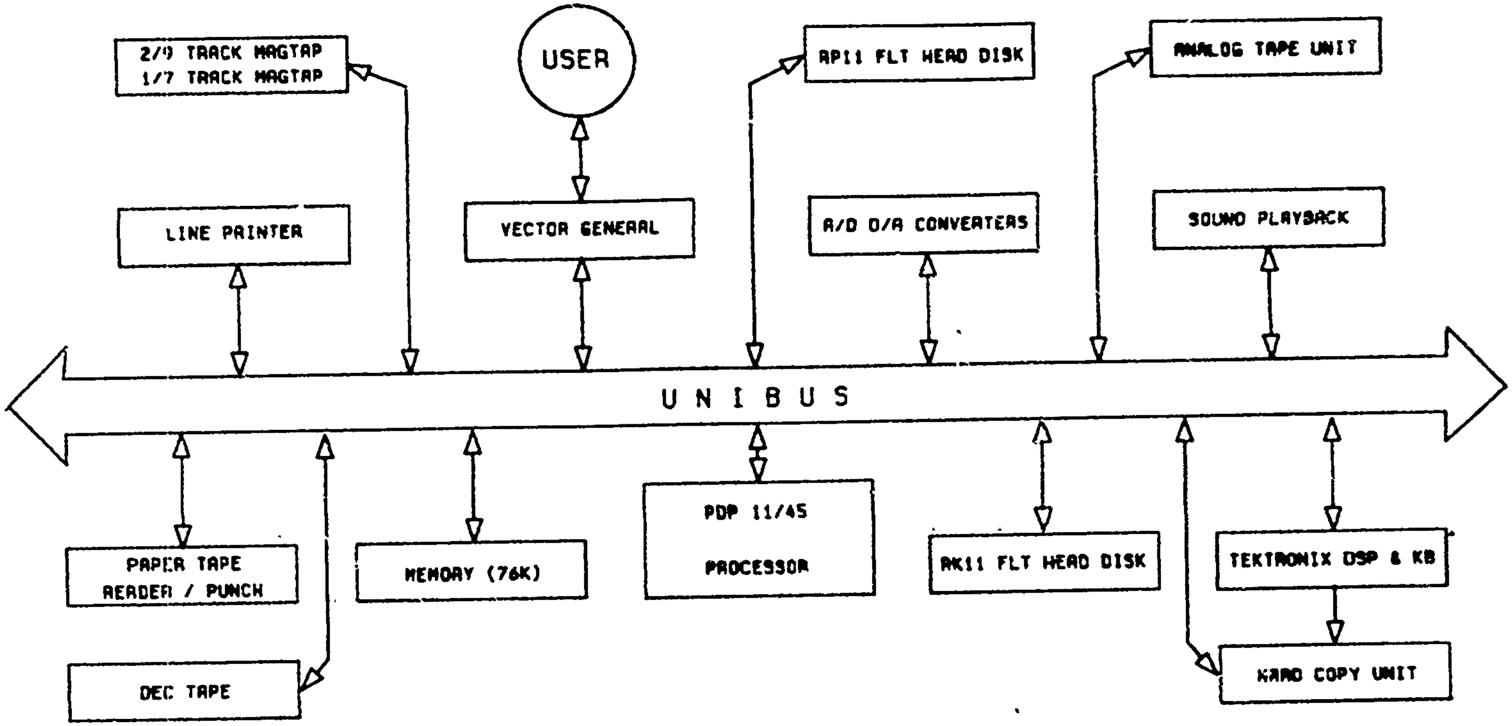


Figure 2-2. The hardware layout for WPS

feature requires that only qualified WPS support people effect system expansion. Since WPS is an operating system by itself, the general systems programmer is not likely to be familiar with the WPS file system or general system procedures.

The VG display is refreshed through an 8K word host memory partition. Since the VG is a refresh display, the amount of data which can be displayed is a function of the length of time required to refresh the screen so as not to result in objectional flicker. This restricts both the number of waveform lines per page and, for the OLPARS segment, the number of symbols per display page. Generally, this problem is overcome by breaking up the data to be displayed into several pages.

The feature extraction language PARLAN is described in detail elsewhere. PARLAN and the associated predicates are the only means of implementing the user's feature extraction algorithms. PARLAN deviates from the Fortran constructs in the area of generalized I/O and data type declarations. The principle data types of PARLAN are "vector" and "waveform." Some support is included for integer and floating point variables. Both "vector" and "waveform" data types are utilized as subscripted arrays. The sophistication of PARLAN comes from its ability to accept a waveform tree as input and generate a transformed waveform (or vector) tree as output. PARLAN can also perform polymorphic transformations; that is, one waveform (or vector) tree as input generating multiple waveform (or vector) trees for output.

PARLAN programs must be inputted into WPS on punched cards. A reasonable amount of compilation error messages are provided, but the complexity of run-time error analysis is beyond the scope of the system.

The linkage between WPS and the user's PARLAN program is accomplished by selecting the option "Execute Program." The user must identify the type and value for each input and output variable. This process is required since WPS itself is not PARLAN structured data. Data trees are interpreted by the system and processed sequentially through the user program. This procedure can become time-consuming since every run of the user algorithm requires reestablishment of the WPS to PARLAN linkage query session.

2.2.2. Macroscopic View of the WAVES Software

The WAVES system is actually a collection of routines invoked by a name assigned by the MULTICS operating system to perform a specified option. The assigned name is actually a command (e.g., list_trees). The command parameter defaults are retrieved from a common region, parts of which are allocated to the various commands. All of the WAVES options are programmed. Each of these options is implemented in the form of a subroutine. Generally, each command has an equivalent subroutine call which can be part of a user-written task. Further, all the primitives of the WAVES system are carefully structured and documented for incorporation into user-written tasks. Since MULTICS is really the executive of the WAVES system, the system is expandable and permits easy inclusion of user-written routines.

The WAVES system utilizes a Tektronix type terminal which, over phone lines, will generally run at 300 baud. Although system response time is good, the time required to display waveforms, restricted by the baud rate, is excessive. The capability of remote access to the WAVES system is made possible by the development of the Remote Data Entry system which enables remote users to transfer data trees to the MULTICS WAVES system. High speed display to the WAVES system is of course available at RADC.

2.2.3. Macroscopic View of the IDRS Software

IDRS is structured as an executive and a series of mutually exclusive subroutine segments suitable for implementation under the RSX 11-M overlay capability on the PDP-11 series computers. The VAX/VMS version of the system is implemented by removing the overlay structure and relying on the VMS virtual memory capabilities to handle the extended task size. The executive and most of the subroutine segments of the RSX 11-M version are written in Fortran. All of the VMS version is in Fortran. Because of the requirement of maintaining certain task size limits on the PDP-11 computers, IDRS is not generally regarded as expandable. Also, since the executive is a program segment, the addition of options to the system requires that the IDRS be very well understood.

The flexibility of the IDRS system comes from its capability to act as a transform box; that is waveform in, transformed waveform and/or display hardcopies out. Because an entire display page is maintained in temporary storage, the page, after being viewed, can optionally be dumped to disk (or mag tape). Further, once the display page has been calculated, subsections of the page are retrievable from temporary storage without being recalculated. The overall effect is that system turn-around time to process an option is very fast.

A version of IDRS which utilizes the PDP-11 and an FPS-120B array processor is also in use. The high speed parallel processing capability

provided by the array processor makes possible complex data transformations in dramatically reduced time from the non-AP version.

2.2.4. Comparative Analysis From a Macroscopic Software Point of View

Although the WAVES system is implemented on the most sophisticated machine of the three systems, it is not necessarily the fastest. One of the attractive features of the WAVES system is its accessibility through the ARPA computer network. For many users, however, low speed modems provide the interface. The high density waveform plots often desired can take several minutes to materialize. The WPS single user system responds with acceptable speed for most applications. IDRS, for normal signal processing investigation, is acceptably quick. For complex signal processing of very large datas, the IDRS configured with a high speed array processor should be considered.

Data entry into the WAVES system has recently been enhanced through the development of the MULTICS Remote Data Entry Capability. Waveform and vector data can be entered into the system without the requirement of shipping data and arranging for its introduction into the WAVES system. Since IDRS and WPS are single user systems, data entry is accomplished locally. WPS data is input from special formatted mag tape or cards. The data must be created out of the WPS environment. IDRS runs in a multi-user, multi-task environment. Data can be accessed from or transferred between any peripherals in the system. An analogue to digital converter, analog tape units, automatic tape search equipment, and digital to analog playback equipment complement IDRS to form a complete signal processing system. The WPS system is also complemented by digital to analog playback capabilities.

Data manipulation in IDRS is accomplished in a straightforward way using the host operating system utilities. Both WPS and WAVES contain a multi-waveform tree capability integral to the system. Both systems supply a wide range of tree manipulation routines. In WAVES, these are available to the user through subroutine calls. In WPS, the user does not have direct access to the data manipulation routines, but rather must rely on the WPS to PARLAN interface.

WPS and WAVES provide a direct linkage to a higher level language for feature extraction. Under WAVES, the user can implement his algorithms in either Fortran or PL/I and normal MULTICS program development aids are available. Since MULTICS is the executive, the user task can be requested just as any WAVES segments are requested. WAVES system subroutine calls manipulate data and, in general, effect any option normally available to the user while at command level. The current data "stack" enables standard input/standard output capability.

In WPS, user algorithms are entered off-line in PARLAN and brought into the system on punched cards. Program execution and compilation are accomplished as WPS options. Data manipulation is performed by WPS with a high degree of user interaction required to accomplish the linkage between the WPS executive and the user PARLAN task.

In the case of IDRS, user algorithms are developed completely external to IDRS, using normal system program development. The user

feature extraction program typically operates on transformed output from IDRS.

The expandability of the various systems is largely a function of structure. WAVES, because MULTICS is the executive which references segments as instructed by the user command, is quite expandable. Additional segments can easily be incorporated. IDRS, written in Fortran, is generally expandable. The VMS version has no memory restrictions, whereas the RSX 11-M version supports a separate overlay for each major program segment. A user should be familiar with this overlay structure before expansion is attempted. Under WPS, expansion should only be attempted by qualified WPS service personnel. The assembly language software and configuration dependent structure and executive is by far the most complex of the three systems.

The WPS must be run on the specified hardware configuration; it is not transportable. In order to run WAVES, a Tektronix-type graphic terminal and access to MULTICS or the ARPA computer network are required. IDRS runs on PDP-11's and VAX's. The high speed version utilizes a PDP-11 and a FPS-120B array processor configuration. A Tektronix 4014 style terminal is recommended for the IDRS graphic terminal.

BIBILIOGRAPHY (Section 2)

1. RADC-TR-78-44, March 1978, Interactive Digital Receiver Simulator (IDRS) System User's Manual and Software Documentation by Pattern Analysis and Recognition Corp. (A053684)

2. PAR #F30602-78-C-0174, Vol. 2, October 1979, Interactive Digital Receiver Simulator (IDRS) System User's Manual [for RADC's Sigint Support Capability] by Pattern Analysis and Recognition Corp.

3. RADC-TR-76-224, Vol. 1 through Vol. 4, October 1976, Waveform Processing System (WPS) by Pattern Analysis and Recognition Corp. Vol I, (A034417) Vol III (A029202), Vol IV, Pt 1 (A038336), Vol IV, Pt 2 (A038330) Vol IV, Pt 3 (A038331)

4. RADC-TR-78-218, Vol. 1 & 2, October 1978, MULTICS Long Waveform Analysis System by Pattern Analysis and Recognition Corp.

5. The RADC Interactive Laboratory for Design of Pattern Recognition Systems and its Applications by

John C. Faust
Haywood E. Webb, Jr.
RADC, Griffiss AFB, NY

Lester A. Gerhardt
Rensselaer Polytechnic Inst.
Troy, NY

SECTION 3

A COMPARATIVE ANALYSIS OF THE VARIOUS VERSIONS OF OLPARS

Two versions of OLPARS (On-Line Pattern Analysis and Recognition System) are available at RADC. The MULTICS OLPARS Operating System (MOOS) is based on a Honeywell 6180 computer available through the ARPA computer network. WPS OLPARS is an integral segment of the PDP 11/45 based Waveform Processing System (WPS). A third OLPARS, "portable OLPARS" is currently being developed under contract to RADC.

The following sections present discussions of each of the systems followed finally by a comparative analysis of the systems. The discussion of the "portable OLPARS," since it is currently being developed, is based on the limited available documentation.

3.1. WPS OLPARS - DETAILED DESCRIPTION

The WPS OLPARS capability is an integral part of the stand-alone WPS system. The system supports its own executive, files processor, and data manipulation, and is written completely in assembly language.

The control logic of the system is tree structured. In order to get from a lowest node frame to an adjacent lowest node frame, the user must first proceed to the next higher level of the tree structure then down to the desired lower node frame. This strict structure helps to prevent errors in user logic by not permitting the selection of meaningless options. Once a main option overlay is brought in, the user will be queried for additional information required by the system. A menu of possible command options is always displayed for the user.

Command options are selected by the user through lighted function buttons which accompany the Vector General (VG) display. Command option parameters are typically entered with the keyboard while graphic input occurs via the light pen, graphic tablet, and potentiometer box.

Data set selection and manipulation requires that a tree be assigned a 6-character tree name and a 5-character name for each node. The first character of the tree name is the tree plotting symbol while the last character of the node name is the node symbol. The "senior node" of the tree is identified by the tree name minus the first letter.

Once a data is selected as the current data set (through the "select data set" option), it need not be respecified unless it is desired to change the current data set.

Results are presented to the user either on the VG display (and optionally on the Tektronix console only for the purpose of hardcopying) or on the system line printer. Since the VG refreshes its display out of the host memory partition (8K words long) it has a maximum number of points which can be displayed. In cases where the amount of data to be displayed exceeds the display buffer capacity, the display is segmented into "pages" and the user controls stepping through the pages. Many routines provide the option of printing additional information on the system printer.

Due to the extreme complexity of the system, system errors are not uncommon. Many are recoverable through the system control and trap mechanism, while in other cases system restart is necessary. Logic and syntactic errors by the user are generally well handled. The program segments generally verify user interaction and supply appropriate diagnostics in the event of "errors." It is recommended that the user understand the system structure, but it is not required (or recommended) that the user concern himself with the internal workings of the system. Certainly the more experience the user has in pattern recognition and the various transformation algorithms, the more likely he is to succeed in obtaining workable solutions.

Data is input to the system via cards or mag tape. Special formats must be adhered to for either method. The lack of operating system support in WPS makes data input one of the most cumbersome frames. Not much input error checking and recovery is effected (e.g. parity error or bad block).

The WPS system supports an RPØ2 disk (25 mega-byte) for data storage. The user can store all the data he has room for up to the 95 unique plot symbols available. The WPS system handles vectors up to 100 dimensions, except for matrix retotaled computations where memory limitations impose the limit of 50. The capability to divide a data tree up into a design set and a test set is provided by the "create tree" frame option which allows a percentage of the vectors to be put in a new design set data tree.

Multi-level logic with Boolean and Longistic type decisions based on the range or relationship of variables is possible. Other features include multivariate normal Baysian classifiers, Mahalonobis distance, and intermixing of various logic schemes within a logic tree. User modifications of logic boundaries and reject regions is generally supported.

It is possible for the user to implement his own algorithms through the use of the PARLAN language. Programs in PARLAN are entered off-line and read into WPS on cards. Editing, Compilation and execution are

performed in WPS. Generally, PARLAN is better suited for feature extraction tasks rather than for computationally intense applications. Batch operation of the WPS system is not provided for. Since the system is single user, it is necessary to share the available resources with all users, unless each has his own mass storage disk. Generally, a user can suspend his work and resume later.

3.2. MOOS - DETAILED DESCRIPTION

MOOS is implemented under the MULTICS operating system written in PL/I. The executive of MOOS is actually MULTICS; that is, user entered commands options are actually requests to run a particular MOOS program. Communication of status and parameters between these programs is accomplished through files stored on disk. MOOS is probably the most powerful and comprehensive version of OLPARS, and, in spite of its complexity, is quite easy to use.

The control structure is single level in the sense that any function may be called at any time, i.e., it is not necessary to back up to a previous level. The ability to call any function at any time sometimes results in logicallv meaningless sequences which are almost always detected, and the user is informed. There is, however, an implied two-level control structure because many major functions have a set of particular options that are meaningful for that major function. (e.g. display manipulation options after projective functions, and ranking options after measurement evaluation functions) Whenever it is reasonable, a menu list of "normal" options is presented to the user; however, he is not restricted to these.

Interaction of the user with the system is primarily via the keyboard with a thumbwheel-controlled cursor on the Tektronix graphic terminal. Selecting a major MOOS function is done, at the command level, by simply entering the program name followed by the treename/nodename

to specify the data set to be operated on. Once the data set has been selected it does not have to be respecified for a subsequent function (unless, of course, the user wishes to change the current data set).

If a function requires further interaction from the user (suboption selection or parameters) the program outputs appropriate prompts to the user.

Presentation of results to the user is flexible and adequate. Many routines have optional output available on the system printer. Logic evaluation output is a summary confusion matrix; more detailed data on those vectors which were misclassified is optionally available. If a confusion matrix is too large for the display screen (>12 classes) it is panelled.

Error detection and recovery is generally well handled. Routines check for many "illegal" conditions and give adequate messages to the user regarding the "error." System errors are relatively few, but not impossible. A major problem regarding system errors is that it is not always obvious that one has occurred, or, sometimes, when it occurred. Sometimes the error doesn't show up until several routines after the routine in which the error occurred.

These cases must be individually handled and experience with the system helps eliminate them and minimize recovery problems. The system

is easy to use and the user doesn't have to be a "systems programmer" to use OLPARS. Of course, the more experience he has in MOOS and in understanding the algorithms he is using, as well as "understanding" his data, the more successful he will be in solving his problem.

Data is input into the system via cards, mag tape, a user-created file, or the remote data entry capability.

The user may have as many trees stored as he wishes (and has space for). He may "work" on up to 20 at one time and has a number of routines available for combining trees and or classes into new trees. Trees and classes can also be deleted, renamed, etc. A test tree may also be created from an existing tree by randomly extracting a (user) specified percentage of vectors.

A variety of projection routines (both 1 and 2 space) are available. Classes may be subdivided, if it is desired, under the structure analysis options.

When designing Fisher logic the user has the choice of eliminating (ignoring) certain specified measurements. Measurement reduction can also be accomplished by creating a new data tree consisting only of select measurements of the original tree.

The user is restricted to a maximum of 100 dimensions and 72 classes in a tree. A limited set of routines which allow measurement evaluation and reduction are available and will allow the user to input a tree with up to 250 dimensions; before he can do normal operations on it it must be reduced to 100 or fewer dimensions. The number of vectors which can be handled (in a class) is a function of dimensionality; i.e. maximum number of vectors (per class) $\leq \frac{64K - 2}{n \dim + 2}$.

A number of algorithms are available for designing logic decision trees. Multi-level logic combining both group and complete-within-group logics is possible. Reject regions are permissible in all logics. Nearest Mean Vector (NMV) and Fisher logic also have modification routines which allow user interaction for "fine tuning" these logics for each class or class pair.

A logic tree or portion of it may be deleted. A logic tree may also be listed on the system printer and it may be converted to a subroutine in FORTRAN source format. The lowest node classnames of a logic tree may also be reassigned new names for the purpose of evaluating a test tree whose class names are different (than the design set).

Batch operation of MOOS is possible through use of the MULTICS "absentee request" capability. If in a user session a problem analysis is not completed the user may save his data or logic for use in a subsequent session. MOOS is a multi-user system and if desired several users may share a data set.

The system may be expanded, but this should be done by a person who is thoroughly familiar with the design and programming conventions.

The user, however, does have the capability of creating a Boolean or linquistic transformation or logic which does allow him a great deal of added flexibility.

3.3. PORTABLE OLPARS - DETAILED DESCRIPTION

The latest version of OLPARS is currently under development for RADC. The advance information presented here is based on available documentation and is therefore far from complete.

Portable OLPARS is written in Fortran and assembly language. All algorithm implementation is written in Fortran. The system interface which comprises some 15% of the system is written in the host computer assembly language. This includes the file access system, display package, and user interface. Portable OLPARS is designed to run on almost any computer with a graphic terminal of the Tektronix 4014 or 4051 type.

The system control structure is a non-rigid tree; that is, the execution of OLPARS programs occurs logically in a tree structural manner, however, the structure is not "enforced" by software. After the execution of an option, a menu of "suggested" options will be displayed. Thus, the tree structure is implicitly implied, but not enforced.

The user's interface to the software system is the Command Input Processor (CIP). This is a system dependent entity which provides a uniform interface to OLPARS for any computer. Abbreviated commands are also accepted.

The system prompts will be in two forms. The user can establish either the "long" or "short" form of command prompt as the default. When the short prompt is active, the user response of "?" will return the long prompt. If the "?" reply is given in response to the long prompt, the system will direct the user to the HELP command for additional aid.

The system design includes extensive error detection and correction mechanisms. The use of a consistent user interface is intended to simplify system operation and standardize system documentation. As with any OLPARS, a general knowledge of pattern recognition is recommended.

The maximum number of dimensions is 50 for the system, 150 while in excess measurement mode. The maximum number of classes is 50 with the total number of vectors in a tree limited to the standard integer size on the host machine (32767 for a 16 bit machine).

The system will not support batch processing, but it is multi-user. One of the primary design features of the system is its expandability. Full support is included for system expansion. The user is provided with documentation illustrating how to incorporate new options into the system.

3.4. COMPARATIVE ANALYSIS OF THE VARIOUS VERSIONS OF OLPARS

When comparing WPS OLPARS, MOOS, and Portable OLPARS, the still under design and development Portable OLPARS cannot fairly be evaluated. The general design and potential of Portable OLPARS match that of MULTICS OLPARS. The design philosophy and structure of the systems is largely the same. The difference is in the executive implementation and expandability. Portable OLPARS is designed to be easily expandable on almost any machine, and to be easy to use because of a consistent user interface. There is every reason to believe that Portable OLPARS will eventually mature to the level of its counterpart, MOOS.

In both MOOS and Portable OLPARS each command directs the system executive to request that a task corresponding to the command option be made active. In WPS, the same command instructs the system to bring the overlay of the command option into core and jump into it. Generally the latter scheme is difficult to expand due to the very complex executive, task size, and memory management considerations. The absence of a high level operating system also restricts hardware modifications or additions since general purpose device drivers are not available.

The filing system and utilities of MOOS and WPS provide essentially a full range of data manipulation options. MOOS is restricted in data storage because of its multi-user environment. WPS provides ample mass

storage capability, but the ability to add peripherals to the system is not present. Portable OLPARS, through its Fortran interface, will have access to all peripherals on the system.

The control logics of the systems are very different. MOOS and Portable OLPARS both support a free structure architecture which permits the calling of any option at any time. WPS, on the other hand, enforces a rigid tree structure which occasionally affects the number of options which must be requested in order to reach a desired goal. Considering the full scope of WPS, however, the implementation of a nonstructured control logic incorporating the WPS waveform processing capability, PARLAN feature extraction language, and OLPARS into an integral system would certainly result in incredibly complex structure. The logical hierarchy imposed lends itself to division into overlay segments for the PDP-11.

The user interaction to the systems is quite different. On the MOOS system (and Portable OLPARS), a persistent graphics terminal with a keyboard and cross-hair capability is utilized. With WPS, a random scan refresh display terminal with keyboard, lighted function switch box, light pen, graphics tablet, and potentiometer box is used. Command option selection in WPS is via the lighted function switches. User query sessions utilize the keyboard. MOOS uses the full or abbreviated command entered at the keyboard. The storage tube graphic display

terminals can display as much data as can be packed onto the screen. The refresh style display, however, is restricted by the size of its refresh memory. Because of this, WPS must page dense displays.

One of the primary points to consider when reviewing these systems for potential use, however, is the data transfer rate to the display terminal. Waveform and to a large degree OLPARS displays, if being sent over a 300 baud modem, can take minutes per display.

Both of the existing systems under investigation strive to supply a flexible and comprehensive pattern recognition capability. Clearly, both of the systems provide excellent service and are well suited for the novice user. The more experience with pattern recognition the user has, the better.

It is not deemed necessary to specifically identify the difference in command options of the various systems, for that can be found in the system documentation. The MOOS documentation is believed to be generally up to date. The WPS documentation is generally quite good, however, it does not reflect the latest system capabilities. For various reasons, several of the options documented are no longer supported. These are cluster plot, nonlinear mapping, and, in the WPS section, pseudo 3-D display, waveform playback, and VG input of PARLAN programs.

Overall, the pattern recognition capabilities provided by both machines are significant. The WPS system, in addition to providing an OLPARS, must provide the services normally associated with an operating system. MOOS, being programmed in PL/I, evolved much faster than WPS, which is entirely in assembly language. The existence of a virtual memory operating system, system primitives, and a higher level language development probably account for the wider complement of options available under MOOS.

BIBLIOGRAPHY (Section 3)

1. RADC-TR-76-271, Vol. 1, Vol. 2, Sept. 76, MULTICS OLPARS Operating System, by Pattern Analysis and Recognition Corp. Vol I (A034393) and Vol II (A033437)

2. RADC-TR-76-224, Vol. I through Vol. II, October 1976, The Waveform Processing System (WPS) by Pattern Analysis and Recognition Corp. Vol I (A034417)

3. MDA904-78-C-0576, January 1980, OLPARS V by Pattern Analysis and Recognition Corporation.

4. MDA904-78-C-0576, November 1979, OLPARS Programmer and System Maintenance Manual by Pattern Analysis and Recognition Corporation.

5. "The RADC Interactive Lab for Design of Pattern Recognition Systems and its Applications."

   John C. Faust
   Haywood E. Webb
   RADC, Griffiss AFB, NY

   Lester A. Gerhardt
   Rensselaer Polytechnic Inst.
   Troy, NY

SECTION 4

FEASIBILITY OF THE MICROPROCESSOR IMPLEMENTATION OF A FEATURE DEFINITION, EXTRACTION, AND ANALYSIS CAPABILITY

The past 20 years of research into pattern recognition problems at RADC has led to the development of the highly-efficient interactive approach which is realized by the WPS, the WAVES, the MOOS, and the IDRS. Each of these systems has repeatedly demonstrated its continuing value and on-going demand. The continued demand for these systems is a tribute to the design philosophy imposed by their creators.

In the on-going effort to better fulfill the needs of pattern recognition analysts, this review has been requested. All of the systems reviewed support the iterative interactive approach outlined here:

1. Waveform Structure Analysis,
2. Waveform Feature Definition,
3. Waveform Feature Extraction,
4. Feature Structure Analysis,
5. Feature Logic Evaluation,
6. Iterate as Required.

Steps (1) and (2) are satisfied by IDRS, WPS and WAVES. WPS and WAVES also provide a mechanism for step (3), whereas this function is performed external to IDRS. Steps (4) and (5) are satisfied by the various versions of OLPARS.

4.1. SYSTEM SOFTWARE REQUIREMENTS

In the analysis of the implementation of a hardware/software system, one must first identify the minimum and optional system capabilities. Experience has shown that the current comprehensive systems contain adequate capabilities for even the most complex problems. What subset of the capabilities can be identified so as to provide a useful, flexible system adaptable to a microprocessor implementation?

We now define the three phases of the process we wish to implement. First, feature definition, including structure analysis and feature definition capabilities. Second, a feature extraction capability which provides flexibility and, since no library of feature extraction routines is adequate for all feature extraction problems, the capability for the user to implement his own feature extraction algorithm. Third, some form of an OLPARS capability which provides the basic structure analysis and logic analysis capabilities. Clearly, these requirements suggest a WPS type system. It is not our intention to suggest that WPS be implemented on a micro, or even that it could, for the system taxes the limits of its host mini-computer.

Since the design and implementation of the current capabilities, the state of available technology has advanced with huge steps. The new 16-bit microprocessors open up new realms of what is possible to implement in a low cost, quick, and useful system. The advances in hardware

have been accompanied by the development of new programming techniques and the availability of high level operating systems.

By reviewing the requirements of the feature definition capability of the proposed system, certain minimum options can be identified. A basic capability must have adequate display capability for efficient time and frequency domain analysis. Maximum flexibility in identifying a waveform segment for examination must be preserved. Waveform segmentation should be provided, together with basic transformations such as digital filtering, FFT, and weighting. Emitter analysis problems suggest the application of demodulation and frequency shift capabilities. These will be considered recommended options. Some of the more advanced options on the list of items not required for the basic configuration would include autocorrelation and crosscorrelation functions.

The feature extraction segment of the system should include some library of algorithms which is expandable to include user-written algorithms. A mechanism for the implementation of user-written algorithms is a requirement. A means of easily interfacing with the OLPARS capability is also required.

The OLPARS capability must include fundamental structure analysis and logic design and evaluation capabilities. Eigenvector projection, Fisher projection, coordinate projection and data set separation on the

basis of projection or Boolean expression seem appropriate requirements for the structure analysis capability. For the logic design and evaluation, simple nearest mean and Fisher discriminant logic with reject strategies seem appropriate.

The system must have adequate histogram and scatterplot capabilities for efficient human analysis of results. A graphic hardcopy is required and, optionally, a line printer capability.

How then can today's technological capabilities and software expertise be best utilized to meet the requirements set forth above, if indeed they can be met? The macroscopic software view of the system will now be considered.

Consider first, the requirements for a mechanism for user feature extraction algorithms. Without a flexible means of accomplishing this, the most powerful feature definition system and OLPARS capability are severely restricted. Experience has shown that this point is a bottleneck in the process of waveform pattern recognition. For these reasons, the support for a higher level language is a requirement. Further, the higher level language should allow maximum transportability, e.g. Fortran or Pascal.

The support for a higher level language implies that the system itself could be written in the higher level language. Further, support for a <u>standard</u> higher level language implies the existence of an operating

system and monitor in order to provide file processing, peripheral communication, and program development support, e.g. editing, compilation, etc. Currently, many microprocessors support both Fortran and Pascal. Development of a system in a higher level industry-compatible language can be done on available large scale computer systems (e.g. MULTICS, VAX/VMS or RSX). This also means that many available software algorithms could be included in the system. Development of the system in a higher level language also implies maintainability and, given efficient system design, expandability.

The application of modular programming techniques should be employed. Three independent segments of the system have been defined. They are mutually exclusive; they will never be needed at the same time. The clever system designer could separate the functions into three separate modes of operation. System command options should invoke separate tasks which communicate system status, options, and parameters via files stored on disk as is done in the MOOS and WAVES system. The results of one option are stored on disk and are inputted by the next sequential option. The three segments can be treated separately, and the options within each section implemented separately.

With this method of design, the system control logic is greatly simplified. The presence of an operating system greatly simplifies system design and development. However, there are still several areas

of complicated system control, for example, if the system could process waveforms by the tree or singularly. The first approach is certainly the most desirable and constant, since the OLPARS requires a tree structure by class. Another example is the desire to average raw power spectra of filtered data segments. This implies a command option looping capability.

Further support for this capability is evident when one examines the logical phases of a problem solution as they relate to system control logic. First, sample waveforms are identified and examined and potential features are decided upon. A feature extraction algorithm is implemented and the user is now ready to pass the entire current data set through the feature definition/feature extraction task pair. The integrated system logic to perform this operation is complex. An alternate approach is to provide a mechanism which optionally has control over the systems. In the same way that IDRS, which is a single waveform capability, can be used to process an entire data tree via the Indirect Command File Processor of the VMS or RSX operating systems, a predefined sequence of command options can be utilized to automatically string together command options. This provides batch processing capabilities. When dealing with the slower computation time of microprocessors, batch processing could be a significant capability.

This concept is not implemented at the system level, but rather it extends control over the system. The concept is presented here as an

alternative which could simplify system design. Some microprocessors are currently supporting the instruction set for a variety of larger machines, for example, a PDP-11. It is now possible to run a high level operating system such as DEC's RT11 on a microprocessor. The RT11 system is ideal for such a system implementation and should certainly be considered.

In summary, an operating system with higher level language support is required to facilitate efficient system design, fast program development, system maintenance, and expandability. Alternatives in system control logic which must be examined include mutually exclusive program segments for feature definition, feature extraction, and OLPARS capabilities. Command options should be requesting that a particular task be run, rather than requesting an overlay segment. Communication between tasks and temporary storage should be in the form of disk files.

4.2. SYSTEM HARDWARE REQUIREMENTS

Based on the previous discussion of system capabilities and design philosophy, it is possible to identify the nature of the hardware configuration required. Of primary importance to the net effectiveness of the system is the user interface. A graphics display capability is required for waveform, scatter, and histogram displays. The analysis of waveform data requires significant disk storage. Additional hardware support items shall also be discussed.

When dealing with a microprocessor based system, one must be careful with excessive processor requirements. The display capability of the system is directly related to this. The requirement of host memory refresh is not viable in microprocessor implementations. Two alternatives are attractive. First, the use of a storage tube display (such as the Tektronix) with a crosshair capability for graphic feedback. Alternately, a random scan refresh display can be used, assuming the [illegible] vides its own memory and is communicated with over a standard interface (for example, EIA at 9600 baud). An important point is the ability for graphic feedback, for example, drawing boundaries or establishing a zoom window.

Another consideration for the hardware configuration is the possibility of including a line printer capability. Clearly the inclusion of a full scale line printer could not be justified. Often, though, large amounts of statistical information are required. Detailed results of logic

evaluations or simple feature vector listings are often required. It may be cost-effective to use a simple graphic display unit (' 'yboard), and provide a high speed printer terminal as the system co... This is an option, but it warrants consideration.

The disk storage requirements of the system are quite large. In order that the system not be storage bound, it is suggested that a secondary storage device of between 20 to 80 mega-bytes be included. In the decision of the size of secondary storage capability, the option of whether or not to include an A/D converter should be considered, since this usually requires contiguous space on disk to digitize at high sample rates.

The inclusion in the system of A/D and D/A converters should be carefully considered. This option makes the system virtually self-contained and a powerful laboratory tool or collection type system. The D/A playback capability permits a wide variety of speech experiments to be investigated with acoustic feedback.

The aspect of speed has not yet been addressed. Clearly, some of the desired options of the system are computationally intense. This suggests that the microprocessor will be CPU bound on many applications. Today's technology offers a viable solution to the problem.

The interface of a one or two board array processing capability could solve the CPU bound problems. Available programmable array processor

capabilities could be utilized to perform computationally complex functions, for example, FFT, matrix operations, frequency shifting, and waveform weighting.

The potential inclusion of specialized support hardware requires an extension of the normal realm of microprocessor capabilities, but, clearly, the system proposed herein for efficient microprocessor implementation is not to be construed as a "normal" microprocessor application. The programmable array processor runs in parallel to the microprocessor, resulting in even more speed enhancements. Several such array processor modules are available. Some include dynamic programming, that is, a desired program is loaded via host software when that capability is desired. High level program development support is also available for some units.

Another optional capability to consider is the possibility of providing a modem link to the comprehensive MULTICS OLPARS capability.

4.3. CANDIDATE HARDWARE CONFIGURATION

The hardware configuration chosen must be determined with direct consideration of the software system to be implemented on it. The software dictates certain requirements. This is a waveform data tree processing system. In order that the user be able to maintain the system software, and waveform and vector data, a large disk storage capacity is required. As discussed earlier, computationally intense calculations of the algorithms should be off-loaded to a small array processor in order to maintain system throughput. The requirement for the ability to support user-written feature extraction routines implies support for a program development environment. The associated operating system, program support, and system design suggests the amount of memory required. The nature of the system, stand-alone, suggests that some A/D and/or tape cassette system will be required to transfer data in and out of the system.

Figure 4-1 shows the generalized system diagram. The central processor unit is assumed to be a 16-bit microprocessor supporting 64K words of memory. The microprocessor shares computing requirements with the array processor. The parallel configuration means that a slower, more versatile microprocessor can be utilized.

A candidate array processor board is the CP-132 by CNR, Inc. The CP-132 using 32-bit complex arithmatic (16-bit real, 16-bit complex)

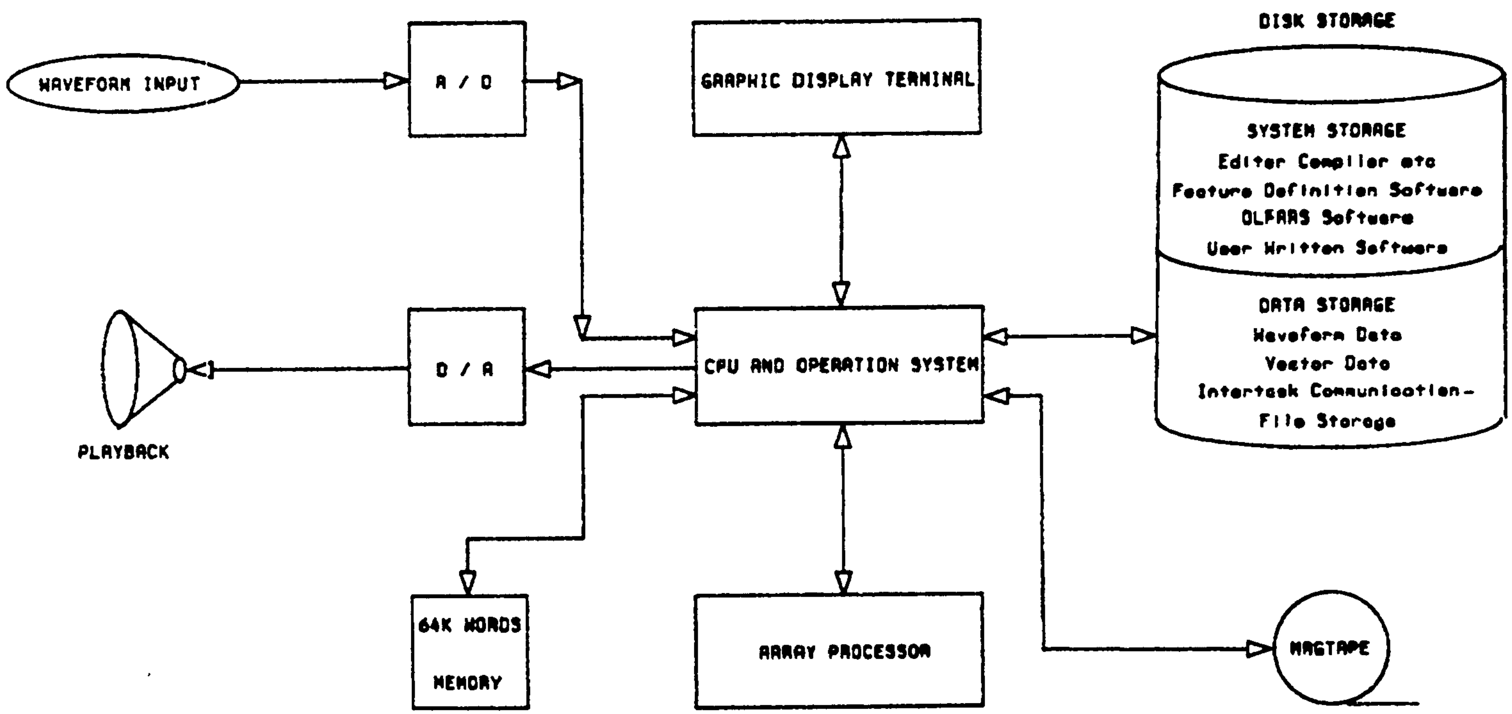


Figure 4-1 Block diagram of the Candidate System Configuration

performs a 1024 point FFT in 1.13 milliseconds. The CP-132 ROM stores signal processing routines which are invoked from the host computer. The unit has 8K x 32 data memory, 4K x 32 table memory ROM, and 2K x 32 program memory RAM. Program development tools for the program source RAM are available. The 16-bit number representation is acceptable for many applications when scaling is carefully controlled. Certainly a more powerful unit with floating point storage is available, though these are generally much more expensive.

A candidate microprocessor is Western Digital Corp.'s MCP-1600. This microprogrammable chip set can be used to emulate the PDP-11 instruction set (it is the basis for the DEC LSI-11 system) and it is interfaceable to the CP-132. The jump up from 32K words of memory requires the inclusion of a memory management unit. This added expense is optional, but recommended, because without it, a task size under an operating system with an I/O page concept is limited to 28K words.

The disk storage capacity must be sufficient; 20 to 40 megabytes would probably be adequate, but certainly the larger, the better. The A/D and D/A units suggested are not very expensive and serve to make a stand-alone capability. The cassette tape provides a means of communication with other systems.

The display requirement will be met by one of the many currently available Tektronix-compatible smaller graphics capabilities. There is

little justification to insist upon a fully expandable display terminal for a specific application such as this.

The hardware configuration candidates discussed here are intended to illustrate the required capability. The microprocessor suggested can support a version of RT-11 (Real-Time Operating System). RT-11 is possibly the most commonly used operating system. It is a very powerful, single user, low overhead, expandable, operating system which supports batch processing, Fortran, Basic, Pascal, and APL. It is intended that the clever system design engineer could configure the candidate system for within the range of 50-55,000 dollars.

4.4. CONCLUSIONS

In the past three sections, we have examined an idea of implementing a waveform feature definition, extraction and analysis system. After analysis of the current systems (WPS, WAVES, and IDRS), certain minimum design criteria became evident.

Effective feature extraction requires the ability to implement user algorithms. In order that the user not require specialist's training, a standard higher level language is required. Implementation of a standard higher order language and the associated program development support (edit, compile, link, etc.) requires (1) a standard file structure, (2) a files processor, and (3) a system monitor. Support for these implies a higher level operating system, such as RT-11. The existence of an effective low overhead operating system and higher level language support suggests that the feature processing system should be developed in a higher level language. This suggests that, indeed, program development for the system can be performed on larger, more powerful machines and the final system installed on the microprocessor based system.

The inclusion of an array processor capability configured in parallel solves many of the problems related to the system response time. The AP is used to perform complex time-consuming algorithms such as the FFT. Certainly this is not a simple microprocessor configuration, but this is not a simple problem.

In the following paragraphs, we shall analyze the feasibility of the candidate system proposed. The evaluation reflects currently available technology as discussed previously.

In any large scale system, one of the key factors to all-around system efficiency is the net system complexity. The system proposed utilizes the powerful RT-11 operating system as a basis. This removes the concerns of file structures, file access, peripheral communication, and hardware expandability from the system. Standard device drivers are used with the exception of the array processor (although system support routines exist). RT-11 also provides for the usage of great amounts of currently available software. Complexity is further reduced by treating the system as three (or more) separate systems, such as

- o data input (if via A/D)
- o feature definition
- o feature extraction
- o OLPARS

Each of these segments in turn is broken down into command options which are implemented as separate tasks (as with WAVES and MOOS). Several options are provided for this under RT-11 including task "spawning," an Indirect Command File capability, and batch capabilities. Notice that different tasks may still share common subroutines. Inter-task communication can be handled either with disk files, memory regions, or both.

With this scheme, not only is maximum flexibility obtained, but the system is completely expandable. In order to include a user-written task in the system (which of course can reference all the data manipulative subroutines as in WAVES), the user need only insert the name of the task in the "menu" capabilities and place it with the rest of the command option tasks. Certainly this design provides for complete flexibility through standardization. Peripherals are equivalent, system language is industry standard, and the user interface through high level language and graphics is flexible.

The accuracy of a microprocessor system is only as good as the way it is programmed. Supporting a PDP-11 instruction set and Fortran, for example, provides full floating point number instruction support. The suggested array processor can be used effectively when appropriate scaling is observed. Of course, the utilization of a floating point array processor (or a floating point processor) is an alternative.

Perhaps the most commonly asked question when considering the implementation of anything on a microprocessor is, "how fast will it run?" This depends on many things. The inclusion of the array processor to off-load some of the more tedious computations is the key to success. The type of system proposed here, with today's technology, has roughly equivalent computing power to the PDP-11/45 WPS system at a fraction of the cost.

In final analysis, the time has again come for the update of our current resources in such a way as to best meet the needs of the pattern recognition sciences. The current technology coupled with experience from earlier successes and system design expertise suggests that a portable and powerful microprocessor implementation of a waveform structure analysis, feature definition, feature extraction, and feature analysis system is already overdue.

BIBLIOGRAPHY (Section 4)

1. The RADC Interactive Laboratory for Design of Pattern Recognition Systems and its Applications, May 1975,

   John C. Faust, Haywood E. Webb, Jr., RADC, Griffiss AFB, NY

   Lester A. Gerhardt, Rensselaer Polytechnic Inst., Troy, NY

2. RADC #TR-76-224, Vol. 1 through Vol. 4, June 1976, The Waveform Processing System by Pattern Analysis and Recognition Corp.
   Vol I (A034417), Vol III (A029202), Vol IV, Pt 1 (A038336)
   Vol IV, Pt 2 (A038330) and Vol IV, Pt 3 (A038331)

3. RADC #TR-76-271, Vol. 1 and Vol. 2, September 1976, MULTICS OLPARS Operating System by Pattern Analysis and Recognition Corp.
   Vol I (A034393) and Vol II (A033437)

4. RADC-TR-78-44, The Interactive Digital Receiver Simulator (IDRS) System Users Manual and Software Documentation by Pattern Analysis and Recognition Corp. (A053684)

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.